

4D Server

Manuel de référence
Windows® / Mac OS®



4D Server® et 4^{ème} Dimension®
© 1985-2006 4D SA. Tous droits réservés.

4D Server - Manuel de référence

Version 2004 pour Windows® et Mac OS®

Copyright © 1985-2006 4D SA
Tous droits réservés

Les informations contenues dans ce manuel peuvent faire l'objet de modifications sans préavis et ne sauraient en aucune manière engager 4D SA. La fourniture du logiciel décrit dans ce manuel est régie par un octroi de licence dont les termes sont précisés par ailleurs dans la licence électronique figurant sur le support du Logiciel et de la Documentation y afférente. Le logiciel et sa Documentation ne peuvent être utilisés, copiés ou reproduits sur quelque support que ce soit et de quelque manière que ce soit, que conformément aux termes de cette licence.

Aucune partie de ce manuel ne peut être reproduite ou recopiée de quelque manière que ce soit, électronique ou mécanique, y compris par photocopie, enregistrement, archivage ou tout autre procédé de stockage, de traitement et de récupération d'informations, pour d'autres buts que l'usage personnel de l'acheteur, et ce exclusivement aux conditions contractuelles, sans la permission explicite de 4D SA.

4D, 4D Draw, 4D Write, 4D View, 4D Insider, 4ème Dimension®, 4D Server ainsi que les logos 4e Dimension et 4D sont des marques enregistrées de 4D SA.

Windows, Windows NT, et Microsoft sont des marques enregistrées de Microsoft Corporation.

Apple, Macintosh, Mac OS, QuickTime sont des marques enregistrées ou des noms commerciaux de Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2006 est un produit de Altura Software, Inc.

ACROBAT © Copyright 1987-2006, Secret Commercial Adobe Systems Inc. Tous droits réservés. ACROBAT est une marque enregistrée d'Adobe Systems Inc.

Tous les autres noms de produits ou appellations sont des marques déposées ou des noms commerciaux appartenant à leurs propriétaires respectifs.

Sommaire

1. Introduction..... 5

Présentation.....	7
Architecture de 4D Server.....	13

2. 4D Server en 10 minutes..... 19

Vérification de l'installation.....	21
Créer une base serveur.....	23
Connexion à la base serveur avec 4D Client.....	26
Définir la structure de la base.....	29
Traitement des données avec 4D Server.....	32
Créer une barre de menus personnalisée.....	37
Travailler simultanément avec plusieurs 4D Client.....	40
4D Server est un serveur Web.....	49

3. Utilisation de 4D Server..... 57

Créer ou ouvrir une base 4D Server.....	59
Quitter 4D Server.....	61
Fenêtre d'administration de 4D Server.....	63
Préférences de configuration.....	68
Préférences de publication.....	71
Crypter les connexions client/serveur.....	73
Transférer des données vers les postes clients.....	76
Mise en place d'un miroir logique.....	77

4. Utilisation de 4D Client..... 81

Connexion à une base 4D Server.....	83
Enregistrer l'accès à la base 4D Server.....	87

5. Menus de 4D Server..... 91

Menu Fichier.....	93
Menu Edition.....	96
Menu Process.....	97
Menu Données.....	101
Menu Web.....	102
Menu Aide.....	103

6. 4D Server et le langage 4D..... 105

4D Server et le langage 4D.....	107
4D Server et les ensembles.....	109
Méthode base Sur démarrage serveur.....	113
Méthode base Sur arrêt serveur.....	114
Méthode base Sur ouverture connexion serveur.....	115
Méthode base Sur fermeture connexion serveur.....	120

7. Procédures stockées..... 121

Procédures stockées.....	123
Inscrire les clients.....	133
Import basé sur les procédures stockées (exemple).....	135
Services basés sur les procédures stockées (exemple).....	138

1

Introduction

4D Server est le serveur de données et d'applications multi plate-forme de 4e Dimension.

Avec 4D Server, vous pouvez créer et utiliser des bases de données multi-utilisateurs ainsi que des applications personnalisées en architecture client/serveur. L'architecture client/serveur indépendante de plate-forme de 4D Server gère des applications pour les 4D Client sur PC Windows et Macintosh. 4D Server est doté de puissants outils de développement et de sécurité des données, est évolutif et se connecte à tous les systèmes d'entreprise.

L'architecture de 4D Server est totalement intégrée : le client et le serveur utilisent une application 4D unique, et les développeurs n'ont pas à concevoir deux développements spécifiques — un pour le serveur et un pour les clients. En outre, 4D Server est un serveur "zéro administration". Il est simple à installer, utiliser, administrer et permet de mettre en place des applications peu coûteuses et rapidement rentables.

4D Server comble le fossé existant entre les systèmes de gestion basés sur le partage de fichiers, lents et peu efficaces, et les hôtes SQL, certes puissants mais complexes à concevoir, à déployer, et à faire évoluer dans le temps. Une application 4D Server peut s'intégrer facilement aux systèmes d'information existants dans les entreprises (tels que Oracle, Sybase, ou tout serveur compatible ODBC) à l'aide des plug-ins 4D de connectivité.

Architecture client/serveur intégrée

Un système 4D Server fonctionne avec une application unique pour le serveur et le client. Le logiciel client et l'application serveur sont les composants d'un seul produit, 4e Dimension. L'application 4D Server se compose de deux éléments : 4D Server et 4D Client, qui forment l'architecture client/serveur.

La partie 4D Server réside sur la machine serveur. Elle stocke et gère la base de données sur le serveur et permet aux utilisateurs de manipuler la base à partir de leur propre machine (la machine cliente — ou poste client).

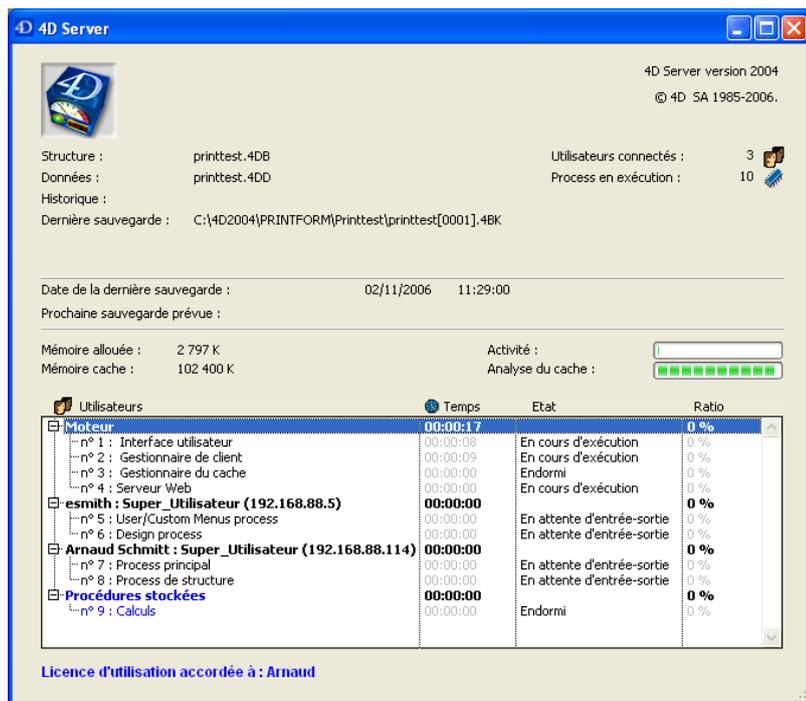
Une copie de 4D Client réside sur chaque machine cliente. Les utilisateurs s'en servent pour accéder à la base de données sur le serveur, et exécuter des opérations de base de données : ajouter de données, créer des états, ou encore modifier la structure de la base de données. Tout ce qui peut être réalisé avec 4e Dimension est également faisable avec 4D Server et 4D Client.

Aucun middleware ni développement supplémentaires ne sont nécessaires pour fonctionner en environnement client/serveur. 4e Dimension, 4D Client et 4D Server partagent les mêmes outils d'interface, le même langage ainsi que le même système de gestion de l'information.

Une application monoposte évolue facilement vers une configuration client/serveur pour le groupe de travail. Réciproquement, lorsque cette application est développée avec 4D Server, elle est de fait une application monoposte avec 4e Dimension.

Serveur de données et d'applications "Zéro Administration"

4D Server est, comme 4D, centré sur l'utilisateur. 4D Server est un système Plug & Play (PnP) complet.



Fenêtre d'administration de 4D Server

- **Administration centralisée en ligne, graphique** : La fenêtre principale de 4D Server affiche automatiquement toutes les informations essentielles concernant l'activité de l'application : mémoire totale allouée à 4D Server, cache de données, nombre et nom des utilisateurs connectés, nombre de process et statut de chacun, activité du serveur, ratio du cache, suivi des sauvegardes.
- **Auto-configurable et évolutif** : 4D Server est conçu pour intégrer de nouveaux protocoles, clients, plug-ins, et pour s'interfacer avec d'autres systèmes, sans qu'il soit nécessaire de le reconfigurer ou d'en modifier la conception.

- **Mise à jour automatique et dynamique des postes clients, contrôle de versions** : Tous les 4D Client reçoivent automatiquement et de manière dynamique les nouveaux éléments d'une application chaque fois que la base évolue ou qu'un plug-in est ajouté, supprimé ou modifié.
- **Connexions automatiques asynchrones via le protocole standard TCP/IP** : 4D Server et 4D Client communiquent de manière transparente via le protocole réseau TCP/IP, quelle que soit la plate-forme du client et du serveur. Le protocole TCP/IP étant aujourd'hui intégré à tous les systèmes d'exploitation, son utilisation ne nécessite aucune installation particulière.
- **Maintenance automatique des sessions et des contextes des connexions 4D Client et Web** : Le moteur de base de données de 4D Server crée et maintient automatiquement un environnement de travail courant pour chaque combinaison table/process/utilisateur. Cette architecture basée sur les sessions permet à chaque process utilisateur de manipuler les données indépendamment et simultanément. En mode contextuel, le serveur Web de 4D Server apporte aux clients Web toutes les fonctions client/serveur.
- **Verrouillage automatique des enregistrements** : 4D verrouille et libère automatiquement les enregistrements. Cette fonction élimine les problèmes courants associés à la modification des enregistrements "en utilisation". Le verrouillage des enregistrements rend caducs les problèmes associés au verrouillage des pages ou fichiers qui sont généralement rencontrés dans d'autres systèmes.
- **Système de messagerie utilisateur intégrée** : 4D Server est né de la micro-informatique. Son interface utilisateur répond aux besoins des environnements de développement intégrés (IDE) modernes. Par exemple, 4D Server est capable d'informer les clients des actions d'administration en cours : déconnexions, sauvegardes, etc.
- **Méthodes de démarrage et de déconnexion automatisées** : 4D Server appelle automatiquement plusieurs méthodes base répondant à des événements spécifiques : Sur démarrage serveur, Sur arrêt serveur, Sur ouverture connexion serveur, Sur fermeture connexion serveur et Sur connexion Web. Par exemple, la Méthode base Sur démarrage serveur peut initialiser et charger automatiquement tous les objets qui seront utilisés pendant la session.

Un jeu de fonctions inégalées

En plus des fonctionnalités de 4e Dimension, 4D Server possède les caractéristiques suivantes :

- **Gestion des données en environnement multi-utilisateurs** : Plusieurs utilisateurs peuvent simultanément effectuer des opérations de bases de données sur la même table ou sur des tables différentes : ajouter, modifier, supprimer, rechercher, trier et imprimer des enregistrements. L'intégrité des données est assurée par un système interne de verrouillage des enregistrements.

- **Développement multi-utilisateurs** : Plusieurs utilisateurs peuvent simultanément développer et concevoir une base de données. Par exemple, les membres d'une équipe peuvent, en même temps, modifier les attributs d'une table, créer et modifier des formulaires et des méthodes. L'intégrité de la structure est protégée grâce à un système interne de verrouillage des objets.
- **Architecture client/serveur indépendante de plate-forme** : Cette architecture gère les performances de la base de données de manière identique pour les clients Windows et Mac OS, que ce soit en matière de multi-développement cross-plate-forme simultanée ou pour les opérations de saisie et de modification de données par des postes clients sur des environnements matériels différents.
- **Architecture de plug-ins Windows et Mac OS** : La version Windows et la version Mac OS de 4D Server permettent d'installer à la fois des plug-ins Windows et Mac OS sur le poste serveur. Cette architecture simplifie la distribution de plug-ins 4D indépendants de plates-formes : ils sont traités de manière transparente par 4D Client et 4D Server, quelle que soit la plate-forme d'exécution du client.
- **Serveur Web intégré** : Tout comme 4e Dimension, 4D Server et chaque 4D Client possèdent un moteur Web qui vous permet de publier les bases 4D sur le Web. Votre base de données peut être directement publiée sur le Web. Vous n'avez pas besoin de développer un système de base de données, un site Web ou une interface CGI entre eux. Votre base de données est votre site Web. Vous pouvez également transformer tout poste 4D Client en serveur Web. Pour plus d'informations concernant le moteur Web intégré de 4D Server et 4D Client, reportez-vous au chapitre Présentation du serveur Web dans le manuel Langage de 4e Dimension.
- **Sécurité des connexions via SSL** (Secured Socket Layer) : Vous pouvez configurer votre 4D Server de manière à ce que les connexions client/serveur s'effectuent en mode sécurisé, par l'intermédiaire du protocole SSL. Pour plus d'informations les connexions client/serveur sécurisées, reportez-vous à la section Crypter les connexions client/serveur.
- **Triggers** : Un trigger est une méthode associée à une table. C'est une des propriétés de la table. Vous n'appellez pas un trigger, il est automatiquement appelé par le moteur de la base de données chaque fois que vous manipulez des enregistrements (ajout, suppression, modification et chargement). Les triggers peuvent être très simples ou très complexes. Vous pouvez les utiliser pour éviter des opérations "illégalles" sur votre base de données. Les triggers représentent une manière très puissante de limiter les opérations sur une table, ou d'empêcher des pertes ou altérations accidentelles de données. Par exemple, vous pouvez, dans un système de facturation, empêcher quiconque de créer une facture qui ne comporterait pas le nom du client. Avec 4D Server, les triggers sont exécutés sur le poste serveur. Tout client, qu'il soit 4D Client ou connecté via 4D Open, est assujéti aux règles de la base de données contrôlées par les triggers. Pour plus d'informations sur les triggers, reportez-vous à la section Présentation des triggers dans le manuel Langage de 4e Dimension.

- **Procédures stockées** : Vous pouvez écrire des méthodes 4D qui seront exécutées en local dans leur propre process sur la machine serveur ou sur un ou plusieurs postes clients que vous désignerez. Ces fonctions sont appelées les procédures stockées, pour utiliser une terminologie couramment employée dans l'industrie du client/serveur. Cependant, 4D possède une architecture qui dépasse le concept standard de procédures stockées. Avec 4D Server, une procédure stockée est en réalité un process serveur (ou un process client, cf. paragraphe suivant) qui exécute le code de manière asynchrone, et indépendamment de tous les autres process exécutés sur les postes client ou serveur. Dans une architecture client/serveur standard, une procédure stockée exécute et retourne (de manière synchrone ou asynchrone) un résultat. Avec 4D Server, vous pouvez démarrer une procédure stockée qui s'exécute pendant toute une session client/serveur, et qui répond, à la demande, aux messages qui lui sont transmis par les clients. Simultanément, vous pouvez lancer une autre procédure stockée qui n'aura aucune interaction avec les clients, mais qui se chargera de synchroniser les données avec un serveur SQL ou un autre 4D Server, via un plug-in de connectivité 4D ou 4D Open. La seule limite au nombre de procédures stockées exécutables simultanément est celle de votre configuration matérielle. Une procédure stockée s'exécute dans son propre process. Donc, comme chaque process utilisateur, elle maintient son propre contexte de base de données (sélection courante). En outre, le langage de 4D possède des commandes qui permettent aux process clients (exécutés sur les postes clients) de lire et d'écrire des variables process dans toute procédure stockée (y compris les variables BLOB). Elles peuvent donc servir à mettre en place un système de communication efficace et puissant entre les clients et les procédures stockées. Avec les procédures stockées, vous pouvez ajouter des nouveaux services personnalisés dans 4D Server. Pour plus d'informations, reportez-vous à la section Procédures stockées.

- **Procédures stockées exécutées sur client** : 4D Server vous permet, à partir d'un poste client ou du serveur, d'exécuter des procédures stockées sur un ou plusieurs autres postes clients. Vous pouvez ainsi optimiser la répartition des charges de travail entre les postes clients et le serveur, ou construire des applications tirant parti des possibilités de communication inter-clients. Pour plus d'informations, reportez-vous à la section Procédures stockées.

- **Chemin d'accès au serveur** : Le chemin d'accès à une base de données serveur peut être enregistré avec un mot de passe utilisateur dans l'éditeur de mots de passes. Cette fonction permet à un utilisateur de se connecter à une base de données sur le serveur en cliquant simplement sur l'icône de document Chemin 4D Client. 4D Client utilise le chemin d'accès pour accéder automatiquement à la base de données souhaitée.

- **4D Open** : 4D Open est l'API (Application Programming Interface) de 4D Server. Elle permet de connecter à 4D Server des applications autres que 4D Client, tournant sous Windows ou Mac OS. 4D Open peut être intégré à 4e Dimension ou à 4D Client et sert à construire des architectures distribuées départementales ou interdépartementales. Enfin, 4D Open utilisé à partir des triggers et des procédures stockées (méthodes exécutées sur le poste serveur) permet de construire des systèmes où 4D Server se connecte à d'autres 4D Server pour effectuer de la réplication ou de la distribution automatique de données. En bref, avec 4D Open, 4D Server peut devenir client de sa propre architecture.

- **Système de sauvegarde intégré** : 4D Server inclut un module complet de sauvegarde des bases de données et de récupération en cas d'incident. Ce module permet de sauvegarder une base en cours d'exploitation, sans qu'il soit nécessaire de quitter l'application. Les sauvegardes peuvent être déclenchées manuellement ou automatiquement, à intervalles réguliers et sans intervention de l'utilisateur. En cas d'incident, la restitution et/ou le redémarrage de la base peuvent également être déclenchés automatiquement.
- **Sauvegarde par miroir logique** : Dans le cadre d'applications critiques, il est possible de mettre en place un système de sauvegarde par miroir logique, permettant un redémarrage instantané en cas d'incident sur la base en exploitation.
- **Plug-ins de connectivité** : Avec les plug-ins de connectivité 4D tels que 4D ODBC Pro, 4D Server et 4D Client accèdent directement à des bases sur mini ou grands systèmes, tels que ORACLE ou toute source de données ODBC. 4D propose également un pilote 4D Server ODBC (Driver ODBC) qui permet à tout client ODBC de se connecter à 4D Server pour travailler avec les données.
- **Connexions distantes** : Dans l'environnement Mac OS, 4D Server supporte les connexions distantes via ARA (Apple Remote Access). Pour plus d'informations, reportez-vous à la documentation d'ARA.
Dans l'environnement Windows, 4D Server supporte les connexions distantes via les **Services d'accès réseau à distance**. Pour plus d'informations, reportez-vous à la documentation de Windows.

Avec son architecture client/serveur, 4D Server ne se contente pas de stocker et de gérer la base de données, mais fournit également des services aux clients. Ces services fonctionnent à travers le réseau par l'intermédiaire d'un système de requêtes et de réponses.

Pour rechercher un ensemble d'enregistrements, par exemple, un poste client envoie une requête au serveur. Dès réception de la requête, 4D Server exécute la recherche en local (c'est-à-dire sur le poste serveur) et, lorsqu'elle est terminée, en retourne le résultat (les enregistrements trouvés).

L'architecture de 4D Server est basée sur le modèle client/serveur. Depuis quelques années, le modèle d'architecture client/serveur s'est imposé, face à son concurrent plus ancien, le partage de fichiers, comme le plus efficace pour les bases de données multi-utilisateurs.

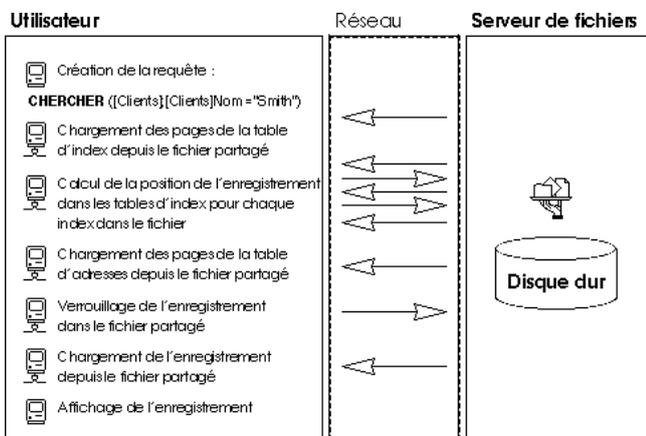
Le type d'architecture client/serveur de 4D Server est comparable à celui qui est utilisée dans le monde de la mini-informatique. De plus, 4D Server apporte deux innovations importantes :

- Une interface intuitive et graphique, présente à tous les niveaux de la base,
- Une architecture intégrée, permettant d'accroître l'efficacité et la vitesse.

L'architecture Partage de fichiers

Avant l'apparition de l'architecture client/serveur, les systèmes multi-utilisateurs exploitaient le partage de fichiers comme modèle d'architecture réseau. Dans ce modèle, tous les utilisateurs partagent les mêmes données mais la gestion des données n'est pas contrôlée par un moteur de base de données central. Chaque poste client doit stocker une copie de la structure et du moteur de la base, tandis que le serveur n'est chargé que de la gestion du logiciel de partage de fichiers sur le réseau.

Dans le modèle du partage de fichiers, chaque station de travail effectue en local toutes les actions de modification sur les données. Cela a pour conséquence de créer un trafic réseau très important, car chaque requête nécessite de nombreuses communications à travers le réseau. Le schéma suivant présente un exemple de trafic réseau généré lorsqu'un utilisateur recherche chaque personne dont le nom est "Smith".



Autre inconvénient du modèle du partage de fichiers : l'impossibilité d'utiliser un cache mémoire pour conserver des enregistrements en mémoire. Si des enregistrements étaient conservés en mémoire, il pourrait exister différentes versions du même enregistrement stocké dans la mémoire cache, ce qui rendrait les données incohérentes. Par conséquent, chaque fois qu'un utilisateur accède à un enregistrement, celui-ci doit être téléchargé depuis le serveur de fichiers. Cela provoque un trafic réseau intense et augmente le temps d'accès aux données.

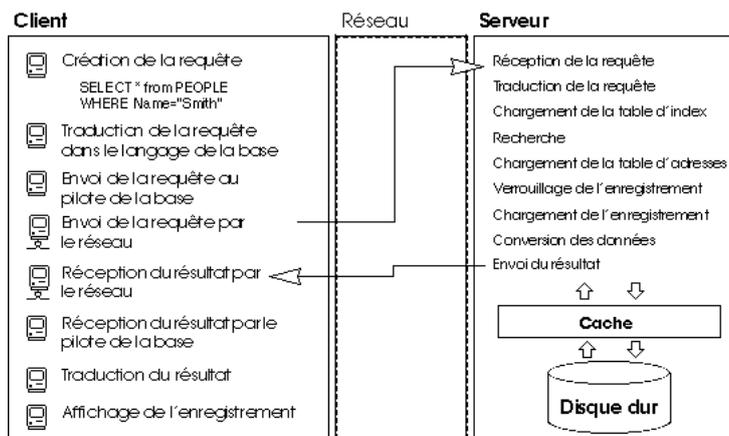
L'architecture client/serveur hétérogène

L'architecture client/serveur est largement répandue dans le monde de la mini-informatique, pour la gestion de bases de données volumineuses, grâce à son efficacité et à sa vitesse. Avec cette architecture, le travail est divisé entre les clients et le serveur de manière à augmenter les performances.

Le serveur contient le moteur central de la base, qui stocke et gère les données. Le moteur de la base est le seul logiciel ayant accès aux données stockées sur le disque. Lorsqu'un client envoie une requête au serveur, le serveur retourne le résultat. Le résultat peut être de toute nature, depuis un simple enregistrement à modifier jusqu'à une liste triée d'enregistrements.

En général, la plupart des architectures client/serveur sont appelées architectures hétérogènes car les applications frontales exécutées sur les postes clients et le moteur de base de données exécuté sur le serveur sont deux produits différents. Dans cette situation, un pilote de base de données est requis pour servir de traducteur entre les clients et le serveur.

Par exemple, pour rechercher un enregistrement, un client envoie une requête au serveur. Comme la base est stockée sur le serveur, celui-ci exécute la commande en local et expédie le résultat au client. Le schéma suivant présente un exemple de trafic réseau généré lorsqu'un utilisateur recherche chaque personne dont le nom est "Smith" et affiche le premier enregistrement trouvé.



Cet exemple illustre deux différences majeures entre le partage de fichiers et l'architecture client/serveur :

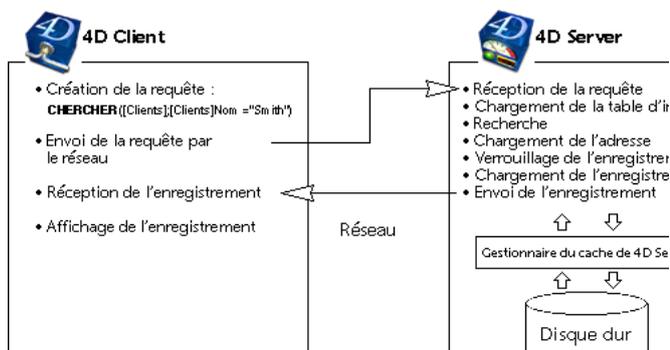
- L'architecture client/serveur autorise l'utilisation d'un cache :** Comme le moteur est le seul logiciel qui dispose de l'accès physique aux données, le serveur peut utiliser un cache qui conserve en mémoire les enregistrements modifiés jusqu'à ce qu'ils soient écrits sur le disque. Comme les données sont envoyées depuis un site central, les postes clients sont assurés de toujours recevoir la dernière version d'un enregistrement. En plus du contrôle de l'intégrité des données qu'il procure, le mécanisme de cache central accélère les opérations de base de données en remplaçant les accès disque par des accès mémoire. Avec le partage de fichiers, tous les accès sont des accès disque.

- Les opérations de base de données de bas niveau sont effectuées sur le serveur :** L'architecture client/serveur permet une augmentation importante de la vitesse d'exécution, car les manipulations de bas niveau sur la base de données, telles que l'examen des tables d'index et d'adresses, sont exécutées localement sur le serveur, à la vitesse de la machine. Avec le partage de fichiers, les mêmes opérations sont ralenties par les transferts sur le réseau et les limites du poste client.

L'architecture client/serveur intégrée de 4D Server

Dans la plupart des architectures client/serveur, l'application cliente et l'application serveur sont deux produits séparés, nécessitant une couche de communication pour pouvoir se comprendre entre eux. Avec 4D Server, l'architecture client/serveur est entièrement intégrée. 4D Server et 4D Client sont deux applications qui partagent la même structure et communiquent directement.

Comme 4D Server et 4D Client parlent la même langue, il est inutile de traduire les requêtes. La division du travail entre le client et le serveur est transparente et est gérée automatiquement par 4D Server.



La division du travail est organisée de telle manière qu'à une requête est associée une réponse. Comme vous pouvez le constater dans le schéma ci-dessus, le client est chargé de traiter les tâches suivantes :

- **Requêtes** : 4D Client envoie des requêtes à 4D Server. Ces requêtes peuvent être construites à l'aide des éditeurs intégrés, tels que l'éditeur de recherches et l'éditeur de tris, ou à l'aide du langage intégré de 4e Dimension. 4D Client dispose d'éditeurs dans lesquels les méthodes peuvent être créées et modifiées. Il gère également les composantes des méthodes telles que les variables et les tableaux.
- **Réception des réponses** : 4D Client reçoit des réponses de 4D Server et en informe l'utilisateur par l'intermédiaire de l'interface utilisateur (des enregistrements différents sont affichés dans un formulaire, etc.). Par exemple, si le client recherche tous les enregistrements dont le nom est "Smith", 4D Client reçoit les enregistrements de 4D Server et les affiche dans un formulaire.

Le serveur est chargé de traiter les tâches suivantes :

- **Gestion des accès** : 4D Server gère toutes les connexions simultanées et les process créés par les clients. Cette gestion tire parti de l'architecture multi-tâche de 4D Server.
- **Objets de structure et de données** : 4D Server stocke et gère tous les objets de structure et de données, y compris les champs, les enregistrements, les formulaires, les méthodes, les barres de menus et les listes.
- **Cache** : 4D Server gère le cache contenant des enregistrements, ainsi que des objets de données créés par les postes clients, tels que les sélections et les ensembles.
- **Opérations de base de données de bas niveau** : 4D Server exécute les opérations de base de données dites "de bas niveau", telles que les recherches et les tris, qui impliquent l'utilisation des tables d'index et d'adresses.

Cette division du travail est extrêmement efficace grâce à l'intégration unique 4D Server et 4D Client. L'intégration de l'architecture de 4D Server est présente à chaque niveau :

- **Au niveau de la requête** : Lorsque 4D Client envoie à 4D Server une requête, telle qu'une recherche ou un tri, 4D Client envoie une description de l'opération de recherche ou de tri en utilisant la même structure interne que 4D Server.
- **Au niveau de la structure et des données** : Lorsque 4D Client et 4D Server échangent un objet de structure ou de données, les deux applications utilisent le même format interne. Lorsque 4D Client a besoin d'un enregistrement, par exemple, 4D Server envoie directement les données dans le format où elles sont stockées sur le disque ou dans le cache mémoire. De la même manière, lorsque 4D Client met à jour un enregistrement et envoie les données à 4D Server, celui-ci les stocke directement dans le cache exactement telles qu'il les a reçues.
- **Au niveau de l'interface utilisateur** : Lorsque 4D Client affiche une liste d'enregistrements, le formulaire utilisé joue un rôle dans l'architecture client/serveur. Par exemple, la fenêtre suivante montre le résultat d'une requête dans la table [Sociétés].

Réf société	Nom société	Téléphone	Fax	Réf ville
ALL	Allinone	+358 9 564		HEL
HYP	Hyperbureau	+33 01 456		PAR
MET	Metaltip	+39 02 789		MIL
ELCO	El Computador	+34 91 147		MAD
NIP	Nippon United	+81 3 258		TOK
GOR	Gorky Town	+00011144		MOS
HUI	Hychen Union	+654 3210		LUX
GOU	Gourdin Industries, Inc	+555 111 5		WAS
GLU	Glurp Technologies	+00 44777		MIL
KOA	Koala Enterprises	+6545454		MEL
BRO	Broceliande	+789456		VAR
KLI	Klick	+666 555 4		MIA

Comme la taille de la fenêtre ne permet d'afficher que douze enregistrements et cinq champs à la fois, 4D Server envoie exactement douze enregistrements. Au lieu d'envoyer la totalité des enregistrements, 4D Server n'envoie que le nombre d'enregistrements et de champs pouvant être affichés dans la fenêtre. Si l'utilisateur fait défiler les enregistrements dans le formulaire, 4D Server envoie les enregistrements et les champs supplémentaires à mesure qu'ils apparaissent dans la fenêtre. Cette optimisation réduit le trafic réseau, car les enregistrements et les champs ne sont envoyés que lorsque c'est nécessaire.

2

4D Server en 10 minutes

Le chapitre d'initiation **4D Server en 10 minutes** vous permet de découvrir rapidement 4D Server. Vous verrez en particulier comment :

- Créer une base serveur
- Connecter un client à la base serveur
- Créer la structure de la base, comprenant des tables, des champs, des formulaires, des menus et des méthodes
- Connecter un second client et travailler simultanément
- Se connecter à la base avec un navigateur Web.

Pour effectuer ces exercices, vous devez disposer d'un ordinateur sur lequel 4D Server est installé, de deux autres ordinateurs sur lesquels 4D Client est installé, et d'un ordinateur sur lequel un navigateur Web est installé.

Avant de commencer à travailler pour la première fois avec 4D Server et 4D Client, nous vous conseillons de vérifier votre installation. Pour cela, lisez la présente section.

Eléments installés

Si vous avez installé 4D Server et 4D Client dans les dossiers proposés par défaut par le programme d'installation, votre disque doit contenir les dossiers et fichiers décrits ci-dessous.

4D Server

Un dossier intitulé 4D Server a été créé dans le dossier d'installation :

- Sous Windows, ce dossier contient l'application 4D Server, ses fichiers associés, ainsi qu'un dossier 4D Extensions. Pour lancer 4D Server, il vous suffit de double-cliquer sur le fichier 4DSERVER.EXE.
- Sous Mac OS, ce dossier contient le progiciel 4D Server. Pour lancer 4D Server, il vous suffit de double-cliquer sur ce progiciel.

4D Client

Un dossier intitulé 4D Client a été créé dans le dossier d'installation :

- Sous Windows, ce dossier contient l'application 4D Client, ses fichiers associés, ainsi qu'un dossier 4D Extensions. Pour lancer 4D Client, il vous suffit de double-cliquer sur le fichier 4DCLIENT.EXE.
- Sous Mac OS, ce dossier contient le progiciel 4D Client. Pour lancer 4D Client, il vous suffit de double-cliquer sur ce progiciel.

Et maintenant...

Notez que le protocole TCP/IP doit être correctement configuré pour que vos machines puissent communiquer à travers le réseau.

Si 4D Server et 4D Client sont correctement installés, continuez avec la section Créer une base serveur. Sinon, si certains des fichiers listés ci-dessus sont manquants, reportez-vous au Guide d'installation de 4e Dimension et procédez à l'installation de ces fichiers.

Cette section décrit la création d'une base serveur à laquelle vous pourrez vous connecter, par l'intermédiaire du réseau, avec 4D Client puis avec un navigateur Web. Si vous utilisez 4D Server et 4D Client pour la première fois, n'hésitez pas en premier lieu à vérifier votre installation. Pour cela, reportez-vous à la section Vérification de l'installation.

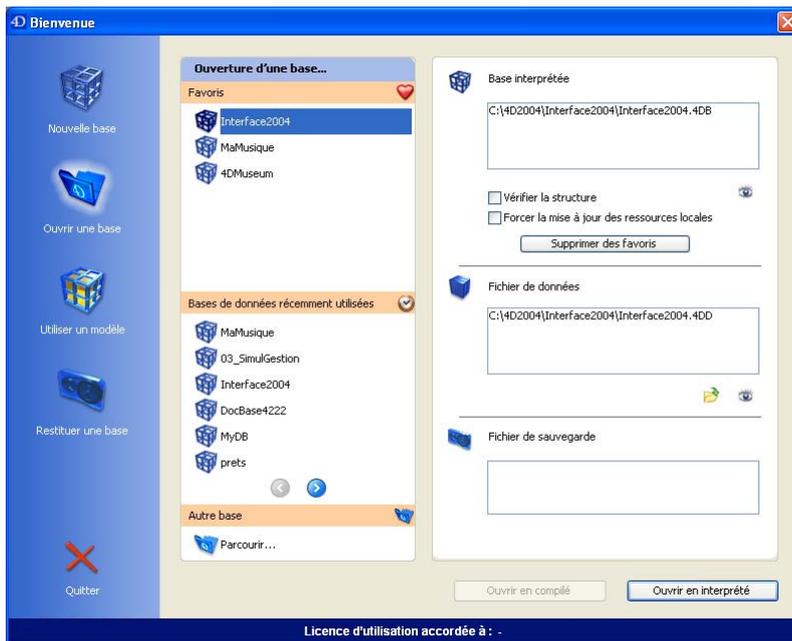
Note : Dans le cadre de cet exemple, nous supposons que vous avez déjà enregistré votre 4D Server auprès de 4D, comme décrit dans le Guide d'installation de 4D Product Line.

Pour créer ou ouvrir une base serveur, lancez 4D Server.

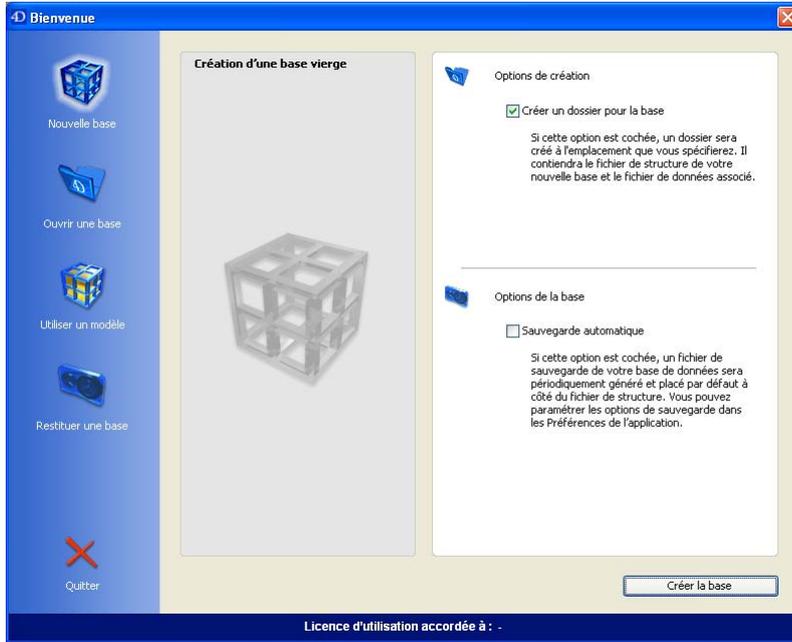
1. Lancez 4D Server en double-cliquant sur l'icône du programme.



La boîte de dialogue de bienvenue apparaît, vous permettant d'ouvrir une base existante, d'en créer une nouvelle (vierge ou basée sur un modèle) ou de restituer une base archivée. Dans cet exercice, nous allons créer une nouvelle base vierge.



2. Cliquez sur l'icône Nouvelle base dans la partie gauche de la fenêtre.
La fenêtre suivante apparaît :



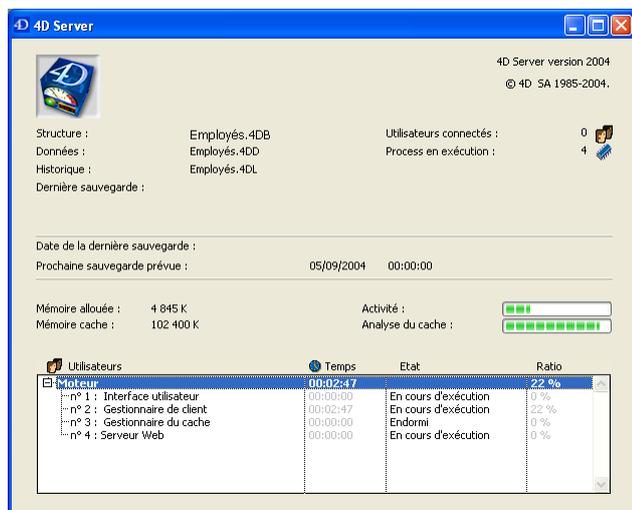
3. Conservez les options par défaut et cliquez sur le bouton Créer la base.

Une boîte de dialogue standard d'enregistrement de fichier apparaît, vous permettant de définir le nom et l'emplacement de la base à créer.

4. Définissez un emplacement puis saisissez le nom de la structure de votre base.

Tapez Employés, puis cliquez sur le bouton Enregistrer.

4D Server crée automatiquement les fichiers et dossiers nécessaires au fonctionnement de la base, puis la fenêtre de gestion des process apparaît :



La **fenêtre d'administration** de 4D Server se compose de plusieurs parties.

La partie supérieure affiche des informations générales sur le serveur et les fichiers utilisés, la partie centrale indique le statut des sauvegardes de la base et les informations relatives à la mémoire, la partie inférieure affiche des informations sur l'activité en cours du serveur.

Notez qu'à cet instant, le nombre d'utilisateurs connectés est zéro. Cela signifie qu'actuellement, aucun client n'est connecté à la base. Le nombre de process en cours d'exécution est de quatre. Ces quatre process sont créés automatiquement par le moteur de la base et le serveur Web intégré.

La partie inférieure de la fenêtre de 4D Server fournit la liste des utilisateurs connectés ainsi que des process en cours d'exécution. Lorsqu'aucun poste client n'est connecté, l'activité du moteur du serveur constitue la seule information affichée.

Et maintenant...

A ce moment, la base est disponible pour les connexions réseau 4D Client Windows et/ou Macintosh. En revanche, la base n'est pas encore prête pour les connexions Web, car ces connexions ne sont pas autorisées par défaut.

Dans cette initiation, nous allons nous connecter dans un premier temps avec 4D Client, définir la structure de la base, puis créer quelques enregistrements. Reportez-vous à la section Connexion à la base serveur avec 4D Client.

Si vous utilisez 4D Server et 4D Client pour la première fois, n'hésitez pas en premier lieu à vérifier votre installation. Pour cela, reportez-vous à la section Vérification de l'installation.

Cette section aborde les sujets suivants :

- Connecter un client à la base serveur que vous avez créée,
- Créer la structure de la base ; la création des tables et des champs, l'ajout et la modification d'enregistrements sont présentés sous forme de petits exercices d'initiation.
- Connecter un second utilisateur,
- Travailler simultanément avec les deux clients.

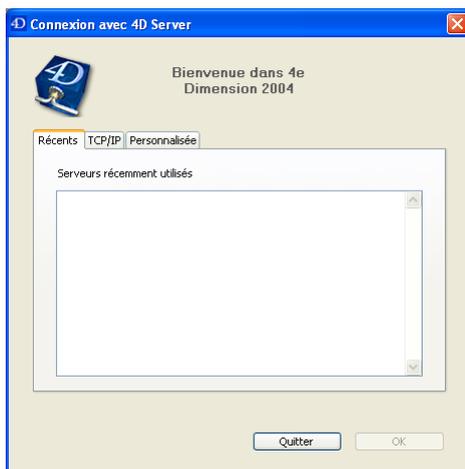
Connexion à la base

Bien que vous ayez créé la base avec 4D Server (cf. section Créer une base serveur), toutes les modifications sur la structure et les données de la base sont réalisées à partir des postes clients. Ce paragraphe vous explique comment connecter un poste client à 4D Server et ouvrir la base serveur.

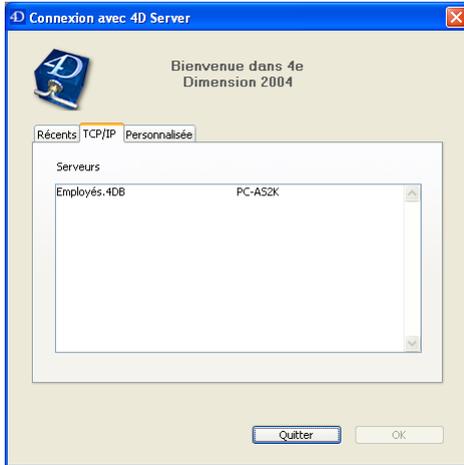
1. Double-cliquez sur l'icône de l'application 4D Client.



La boîte de dialogue de connexion apparaît. Elle affiche par défaut une page listant les serveurs récemment utilisés (la liste est vide lors de la première utilisation).



2. Cliquez sur l'onglet TCP/IP afin d'afficher la liste des bases 4D publiées sur le réseau. La base Employés apparaît dans la liste :



3. Sélectionnez Employés.4DB et cliquez sur OK.

La base s'ouvre sur le poste client, en mode Structure. Vous pouvez dès lors définir sa structure.

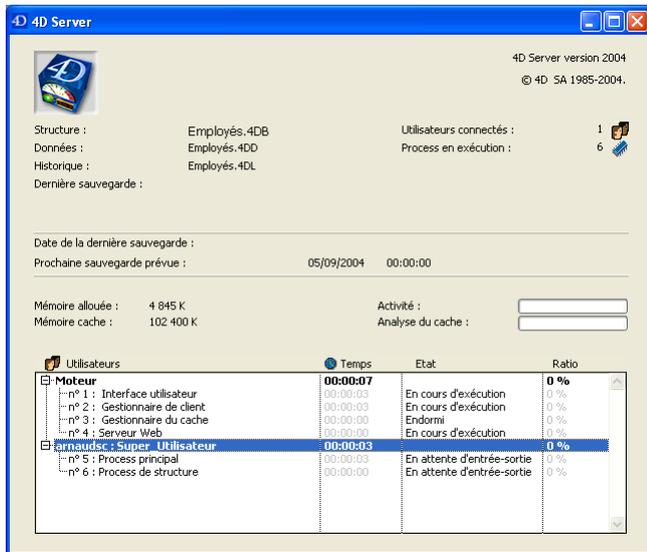
Conseils de dépannage : Si vous ne voyez pas le nom de la base que vous venez de créer avec 4D Server, vérifiez chaque point suivant :

- Est-ce que 4D Server est toujours exécuté sur l'autre poste ?
- Est-ce que les deux machines sont bien connectées au réseau ?
- Est-ce que le protocole réseau TCP/IP est bien configuré sur les deux postes ?
- Pour plus d'informations sur l'utilisation de la boîte de dialogue de connexion, reportez-vous à la section Connexion à une base 4D Server.

Activité du serveur

Si vous observez maintenant la fenêtre d'administration de 4D Server, vous pouvez noter que votre nom d'utilisateur a été ajouté à la liste des Utilisateurs, et que le nombre d'utilisateurs connectés est désormais un.

Il y a maintenant au total six process en cours d'exécution :



Les quatre premiers process sont nécessaires au fonctionnement du serveur et ont été créés au lancement de 4D Server. Les deux nouveaux process sont issus de la connexion du premier utilisateur au serveur :

- Le Process principal gère les modes Utilisation et Menus créés. Chaque utilisateur supplémentaire provoquera la création d'au moins deux process dans la liste.
- Le Process de structure gère le mode Structure. Si vous refermez le process du mode Structure sur le poste client en choisissant **Fermer Structure** depuis le menu **Fichier**, le Process de structure est refermé sur le poste serveur.

Vous pouvez cacher la liste des process d'un utilisateur en cliquant sur le signe "moins" (-) sous Windows ou la flèche sous Mac OS, situé(e) à gauche du nom de l'utilisateur. Pour déployer la liste des process d'un utilisateur, cliquez sur le signe "plus" (+) sous Windows ou la flèche sous Mac OS.

Et maintenant...

Maintenant que vous êtes connecté, vous pouvez travailler avec la base en ayant à votre disposition les mêmes fonctionnalités que si vous utilisiez 4e Dimension version monoposte. Vous devez tout d'abord définir la structure de votre base : reportez-vous à la section Définir la structure de la base.

Une fois que vous vous êtes connecté à la base serveur (cf. section Connexion à la base serveur avec 4D Client), passez la fenêtre de Structure au premier plan.

La nouvelle base comporte par défaut une seule table, nommée [Table1].

Définir la table [Employés]

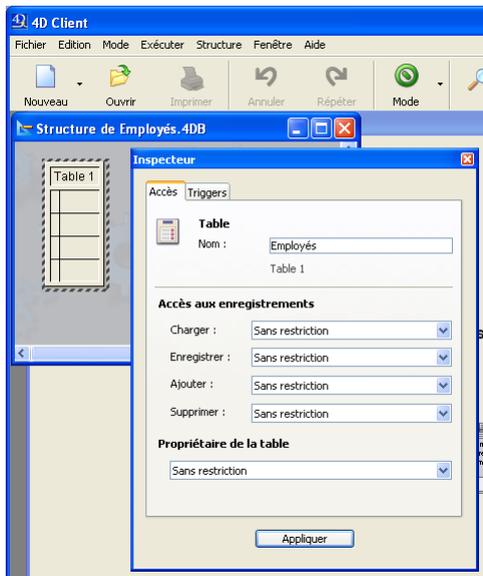
1. Double-cliquez sur la zone de titre de [Table1].

ou

Cliquez avec le bouton droit de la souris ou effectuez Ctrl+clic (sous Mac OS) sur la [Table1] puis choisissez Propriétés de la table... dans le menu contextuel qui apparaît.

La palette flottante Inspecteur apparaît.

2. Saisissez Employés dans la zone Nom de la table.



3. Cliquez sur le bouton Appliquer.

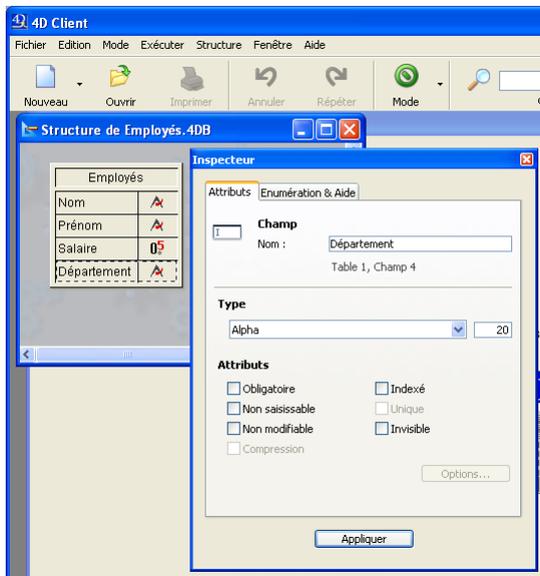
4. Cliquez avec le bouton droit de la souris ou effectuez Ctrl+clic (sous Mac OS) sur la [Table1] puis choisissez Nouveau champ dans le menu contextuel qui apparaît.

Les propriétés des champs s'affichent dans la palette Inspecteur.

4. Créez les champs suivants dans la table [Employés] :

Nom du champ	Type du champ
Nom	Alpha (20 caractères)
Prénom	Alpha (20 caractères)
Salaire	Numérique
Département	Alpha (20 caractères)

Pour chaque champ : saisissez le libellé du champ dans la zone **Nom**, sélectionnez le type de champ puis cliquez sur le bouton **Appliquer**.



Une fois que vous avez créé tous les champs dans la table, refermez la palette Inspecteur.

Note : Si d'autres applications 4D Client travaillent simultanément avec la même base, les champs que vous venez de créer apparaissent sur leurs machines quelques instants plus tard.

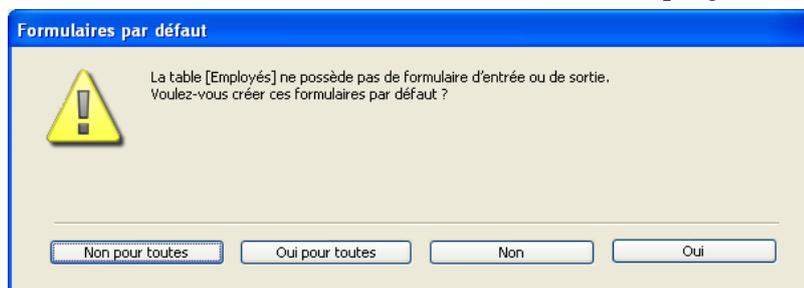
Les modifications sont effectuées sur le serveur en temps réel, mais n'apparaissent pas immédiatement sur les autres écrans afin d'éviter des redessins d'écrans trop fréquents.

Créer des formulaires pour la table [Employés]

Une fois que vous avez défini le contenu de la table [Employés], vous devez créer des formulaires afin de pouvoir saisir et travailler avec des enregistrements pour cette table. Pour cela, vous pouvez utiliser l'**Assistant de création de formulaires** et créer des formulaires comme vous le souhaitez. Cependant, 4D Server vous propose un raccourci pratique pour créer rapidement des formulaires entrée et sortie par défaut.

1. Choisissez Utilisation dans le menu Mode

Vous passez alors en mode Utilisation. 4D Server détecte que la table ne dispose d'aucun formulaire et vous demande si vous voulez laisser le programme les créer pour vous.



2. Cliquez sur le bouton Oui.

La table comporte désormais un formulaire entrée pour l'ajout et l'affichage individuel des enregistrements et un formulaire sortie pour l'affichage ou la saisie de plusieurs enregistrements en mode liste.

Et maintenant...

Votre base serveur est prête pour le traitement des données. Reportez-vous à la section Traitement des données avec 4D Server.

Au cours de la section Définir la structure de la base, vous avez défini les propriétés de la table [Employés] et laissé 4D Server créer les formulaires par défaut pour cette table. Vous êtes désormais prêt pour saisir des enregistrements.

Saisir des enregistrements

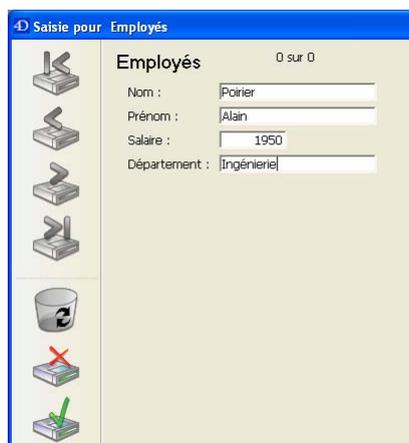
Vous saisissez, recherchez, imprimez et modifiez des enregistrements en mode Utilisation. Si vous ne vous trouvez pas actuellement en mode Utilisation, choisissez **Utilisation** dans le menu **Mode**. 4D Server affiche le formulaire sortie par défaut créé automatiquement, vide pour le moment.

1. Choisissez Nouvel enregistrement dans le menu Enregistrements.

Le formulaire entrée apparaît, vide.

2. Saisissez un premier enregistrement, comme dans l'écran ci-dessous.

Utilisez la touche **Tabulation** ou la souris pour vous déplacer parmi les champs.



2. Cliquez sur le bouton de validation du formulaire pour enregistrer votre saisie.

Un formulaire entrée vide apparaît, vous permettant de créer un autre enregistrement.

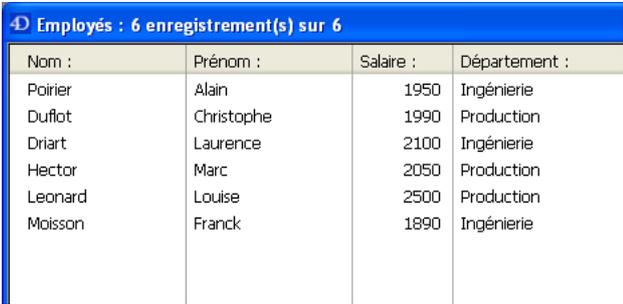
3. Saisissez cinq autres enregistrements avec les valeurs suivantes :

Nom	Prénom	Salaire	Département
Duflot	Christophe	125000	Production
Driart	Laurence	230000	Ingénierie
Hector	Marc	142500	Production
Leonard	Louise	150000	Production
Moisson	Franck	260000	Ingénierie

Une fois que vous avez validé le dernier enregistrement, cliquez sur l'icône d'annulation (comportant une croix), de manière à refermer le nouveau formulaire entrée vide. Le formulaire sortie apparaît.

4. Si les six enregistrements ne sont pas affichés, choisissez Tout montrer dans le menu Enregistrements.

Votre fenêtre doit ressembler à celui-ci :



Nom :	Prénom :	Salaire :	Département :
Poirier	Alain	1950	Ingénierie
Duflot	Christophe	1990	Production
Driart	Laurence	2100	Ingénierie
Hector	Marc	2050	Production
Leonard	Louise	2500	Production
Moisson	Franck	1890	Ingénierie

Les enregistrements sont maintenant stockés dans la base, sur le poste serveur. Si un second 4D Client était connecté au serveur, il pourrait afficher les enregistrements que vous venez de saisir. A l'inverse, si d'autres clients étaient en train de saisir des enregistrements, vous pourriez les afficher en choisissant **Tout montrer** dans le menu **Enregistrements**. Les enregistrements stockés sur le serveur sont accessibles à tous les utilisateurs.

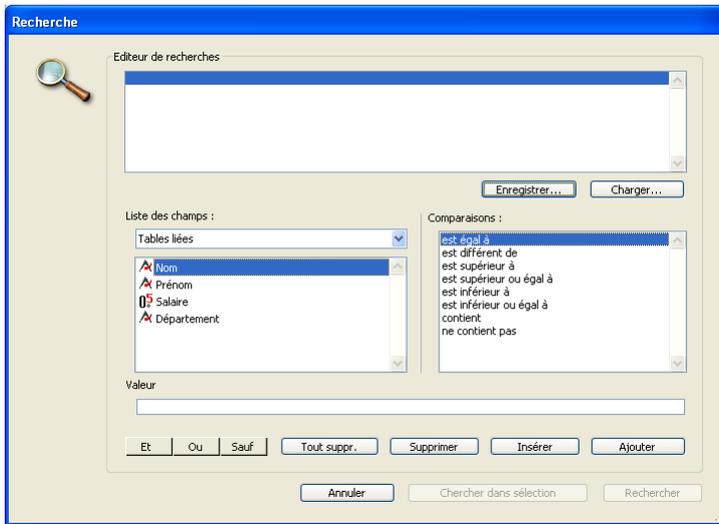
Chercher des enregistrements

Une fois que vous avez créé des enregistrements dans la table [Employés], vous pouvez effectuer des recherches, des tris, des impressions, etc., avec ces enregistrements. Par exemple, recherchons les employés du département Ingénierie :

1. Cliquez sur le bouton Chercher dans la barre d'outils.



L'éditeur de recherches apparaît.



2. Cliquez sur **Département** dans la liste des champs, sur **est égal à** dans la liste des comparateurs, puis saisissez **Ingénierie** dans la zone Valeur.

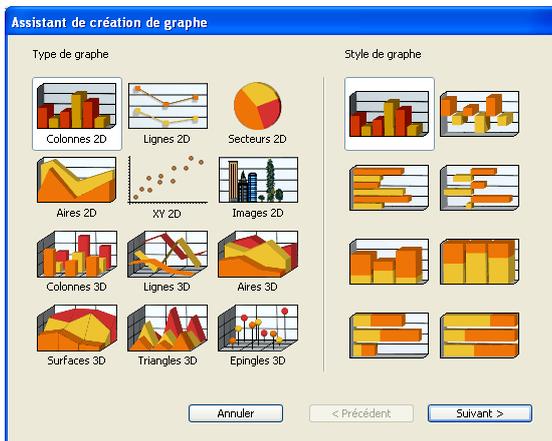
3. Cliquez sur le bouton **Rechercher**.

La requête est envoyée à 4D Server, puis 4D Server répond à 4D Client. Le formulaire sortie affiche alors uniquement les employés travaillant dans le département Ingénierie. Pour afficher tous les enregistrements, choisissez **Tout montrer** dans le menu **Enregistrements**.

Créer un graphique

1. Choisissez **Graphes...** dans le menu **Outils**.

L'assistant de création de graphes apparaît :

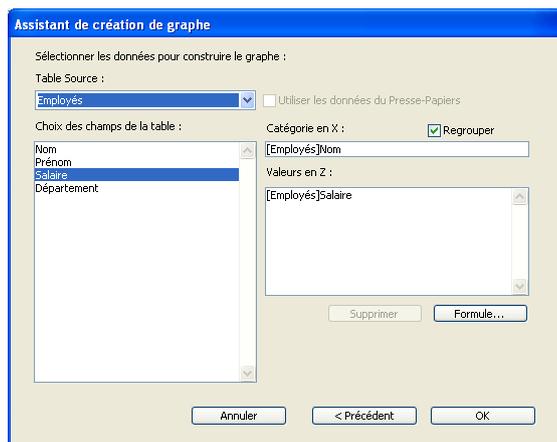


Nous allons créer un graphe en colonnes 2D (ce qui est le type de graphe sélectionné par défaut).

2. Cliquez sur le bouton Suivant >

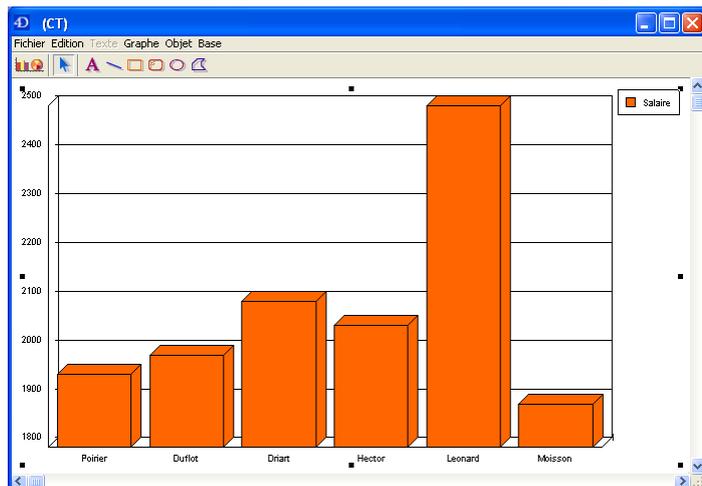
Vous devez sélectionner les données que vous souhaitez représenter graphiquement.

3. Glissez-déposez le champ Nom dans la zone Catégorie en X et le champ Salaire dans la liste Valeurs en Z :



4. Cliquez sur le bouton OK.

Une fenêtre de graphe apparaît, affichant le graphe issu de vos paramètres.



Et maintenant...

En quelques minutes seulement vous avez créé une base serveur, défini une table, ajouté des enregistrements puis effectué une recherche et généré un graphe avec les données saisies dans la base.

Il est maintenant temps de créer une barre de menus personnalisée pour votre base. Reportez-vous à la section Créer une barre de menus personnalisée.

Dans cette section, vous allez créer deux méthodes et une barre de menus personnalisée. En résumé, vous allez créer une application 4D.

Créer les deux méthodes

1. Passez en mode Structure.

Pour cela, choisissez la commande **Structure** dans le menu **Mode** ou cliquez sur une fenêtre du mode Structure.

2. Sélectionnez **Nouveau > Méthode...** dans le menu **Fichier**.

La boîte de dialogue de création de méthode apparaît.

3. Nommez la méthode **"M_AJOUT_ENR"** et cliquez sur **OK**.

Une fenêtre de l'éditeur de méthodes apparaît.

4. Ecrivez le code de la méthode **M_AJOUT_ENR** ci-dessous :



5. Créez une seconde méthode nommée **"M_LISTE_ENR"** avec le code suivant :



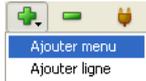
Maintenant que les deux méthodes ont été créées, vous allez définir une barre de menus personnalisée et associer ces méthodes à des commandes de menus.

Créer une barre de menus personnalisée

1. Sélectionnez Boîte à outils > Menus dans le menu Structure.

L'éditeur de barres de menus s'affiche. Il contient une barre de menus par défaut.

2. Placez-vous à la fin de la liste des menus existants (cliquez sur le menu "Mode") et choisissez la commande Ajouter menu dans la partie inférieure de la fenêtre.



3. Saisissez "Initiation" comme libellé de menu et appuyez sur Entrée.

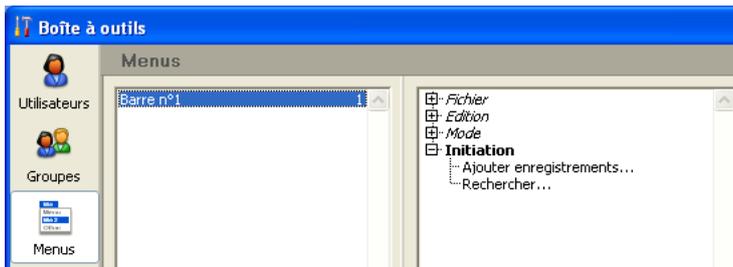
4. Choisissez la commande Ajouter ligne dans la partie inférieure de la fenêtre afin d'ajouter une ligne au menu "Initiation".

5. Saisissez "Ajouter enregistrements..." comme libellé et appuyez sur Entrée.

6. Choisissez la commande Ajouter ligne afin d'ajouter une seconde ligne au menu "Initiation".

7. Saisissez "Rechercher..." comme libellé et appuyez sur Entrée.

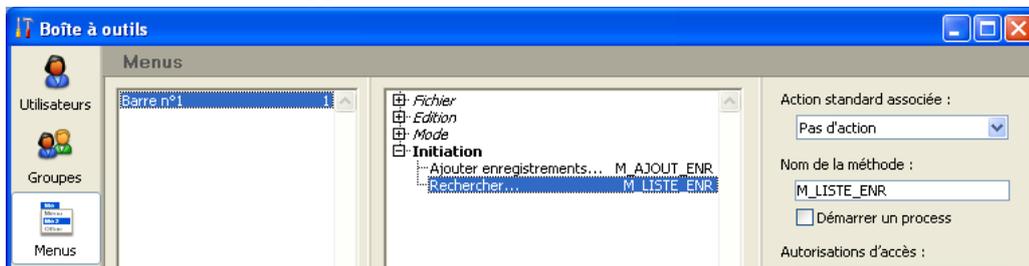
La barre de menus n°1 doit maintenant avoir l'aspect suivant :



8. Cliquez sur l'intitulé "Ajouter enregistrements..." et saisissez "M_AJOUT_ENR" dans la zone Nom de la méthode.

9. Cliquez sur l'intitulé "Rechercher..." et saisissez "M_LISTE_ENR" dans la zone Nom de la méthode.

La barre de menus n°1 apparaît alors ainsi :

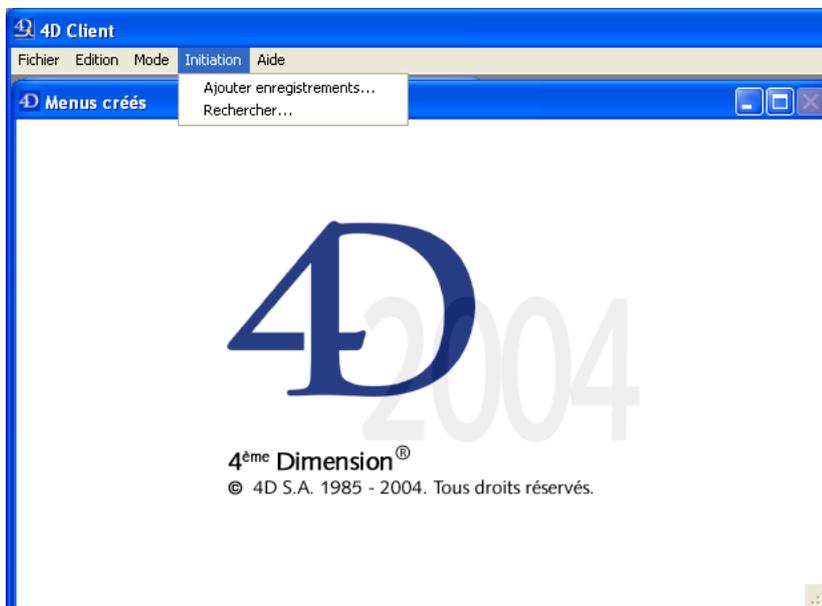


10. Refermez la fenêtre de la boîte à outils.

C'est tout !

11. Sélectionnez **Menus créés** dans le menu **Mode**.

Vous pouvez alors utiliser l'application avec les menus que vous venez de créer :



Si, par exemple, vous sélectionnez **Rechercher...** dans le menu **Initiation**, l'éditeur de recherches de 4D (l'éditeur intégré du mode Utilisation) apparaît. Vous pouvez alors construire votre recherche, puis afficher et modifier les enregistrements trouvés.

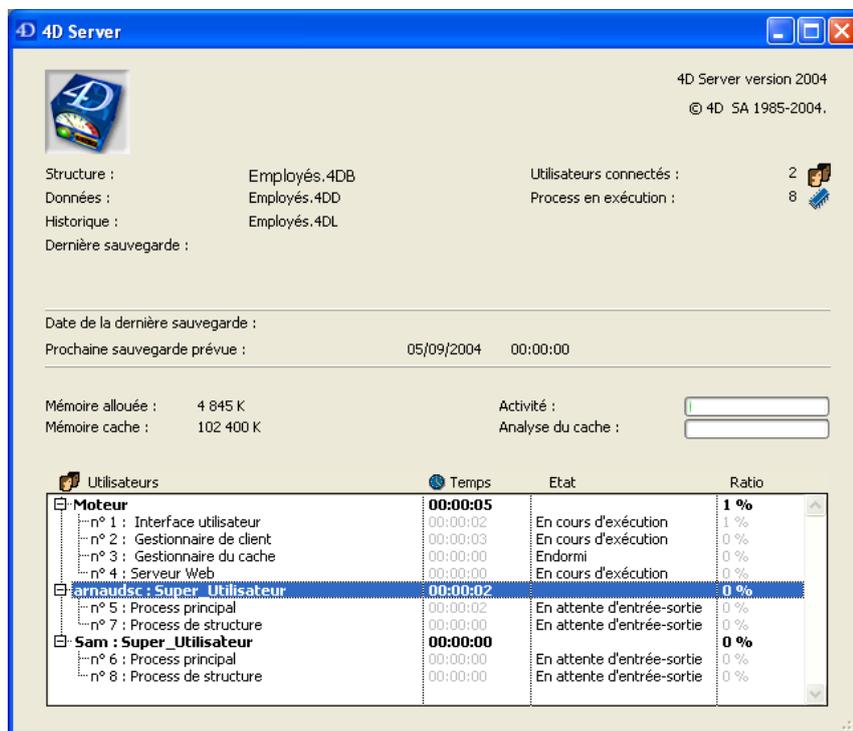
Mais sachez surtout que, sans le savoir, vous venez de développer deux applications !

Pour savoir pourquoi, rendez-vous à la section Travailler simultanément avec plusieurs 4D Client.

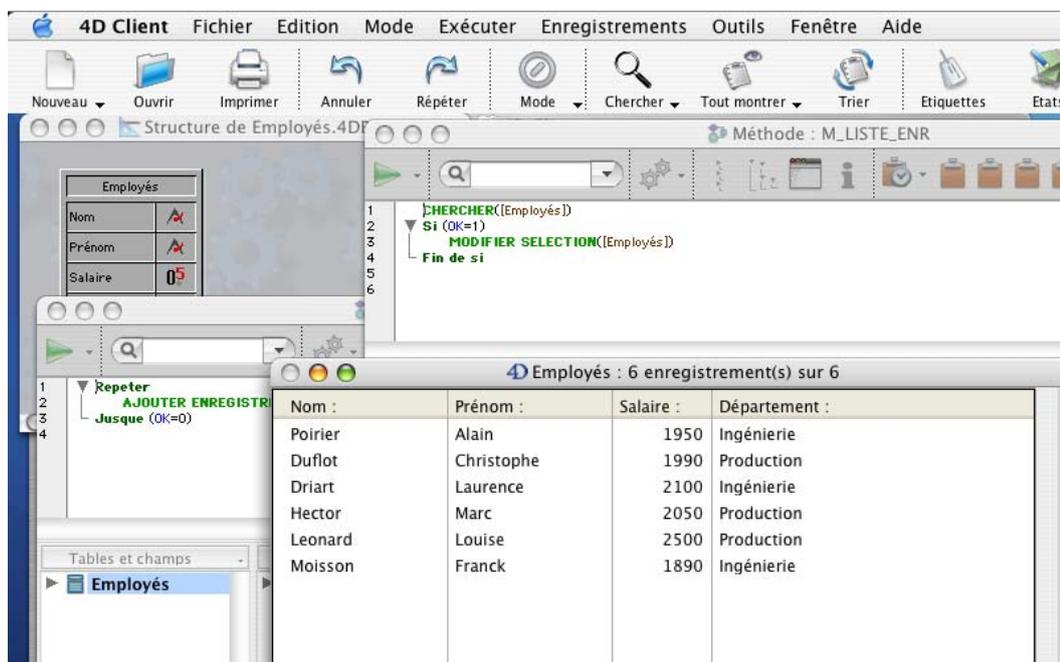
Si vous avez effectué les exercices de cette initiation sous Windows, vous pourriez utiliser la base serveur "telle que" sur Macintosh. Si vous avez effectué les exercices de cette initiation sur Macintosh, vous pourriez utiliser la base serveur "telle que" sous Windows.

Connexion à la base serveur avec un second utilisateur

Pour les besoins de cette initiation, nous nous connectons à la base serveur avec un 4D Client Windows et un 4D Client MacOS. Dès que la connexion est établie, vous pouvez visualiser le second utilisateur dans la fenêtre des process de 4D Server :



Sur chaque poste client, tout ce qui a été effectué sur la base est réutilisable instantanément et de manière transparente. Voici le mode Structure sur un client MacOS :

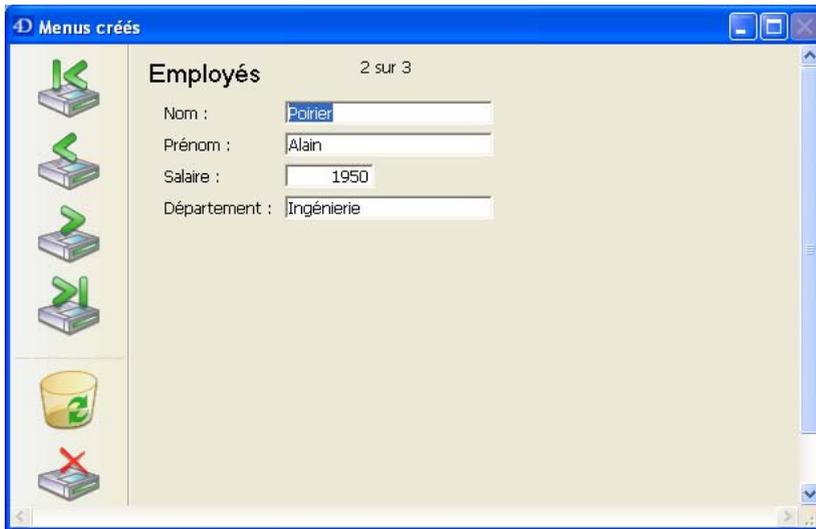


Vos six enregistrements et vos deux méthodes sont là !

Travailler simultanément sur les enregistrements

Sur le premier poste client, passez en Menus créés, choisissez **Rechercher...** dans le menu **Initiation** et recherchez les enregistrements pour lesquels “Département est égal à Ingénierie”. Faites la même chose sur le second poste client. Sur les deux postes, vous obtenez une liste composée de trois enregistrements.

Sur la première machine, double-cliquez sur l'enregistrement "Poirier, Alain". Votre écran affiche la fenêtre suivante :



Faites la même chose sur le second poste. Le système de gestion intégré du verrouillage des données de 4D Server vous avertit que l'enregistrement est déjà en cours d'utilisation :



Vous avez tout de même accès à l'enregistrement en lecture seulement (vous pouvez le visualiser mais pas le modifier) :

The screenshot shows a window titled "Saisie pour Employés" with a sidebar of icons. The main area displays the following form fields:

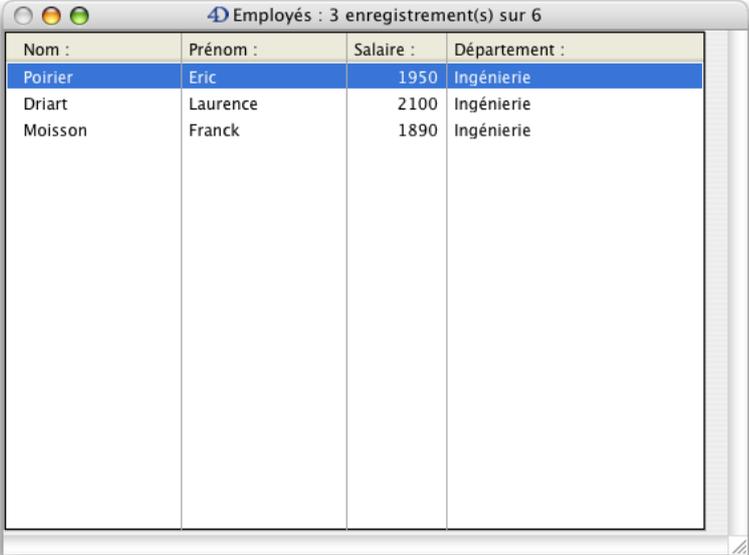
Nom :	Poirier
Prénom :	Alain
Salaire :	1950
Département :	Ingénierie

Sur la première machine, modifiez le prénom en "Eric" et validez votre modification. Vous obtenez :

The screenshot shows a window titled "Menus créés" containing a table with the following data:

Nom :	Prénom :	Salaire :	Département :
Moisson	Franck	1890	Ingénierie
Poirier	Eric	1950	Ingénierie
Driart	Laurence	2100	Ingénierie

La liste a été mise à jour avec le nouveau prénom. Sur la seconde machine, cliquez sur le bouton d'annulation de l'enregistrement dans le formulaire entrée. Vous obtenez :



Nom :	Prénom :	Salaire :	Département :
Poirier	Eric	1950	Ingénierie
Driart	Laurence	2100	Ingénierie
Moisson	Franck	1890	Ingénierie

La liste a également été mise à jour avec le nouveau prénom !

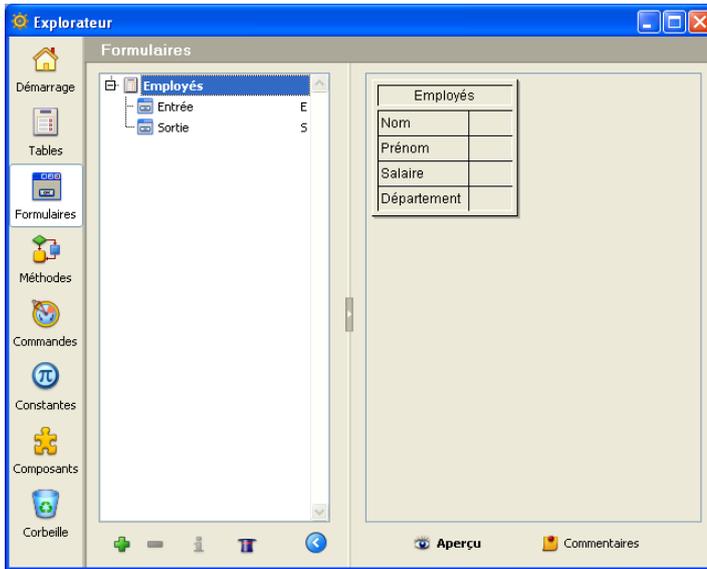
Travailler simultanément avec des objets de structure

4D Server est un serveur de données **et** d'application. Nous allons voir ce que cela signifie.

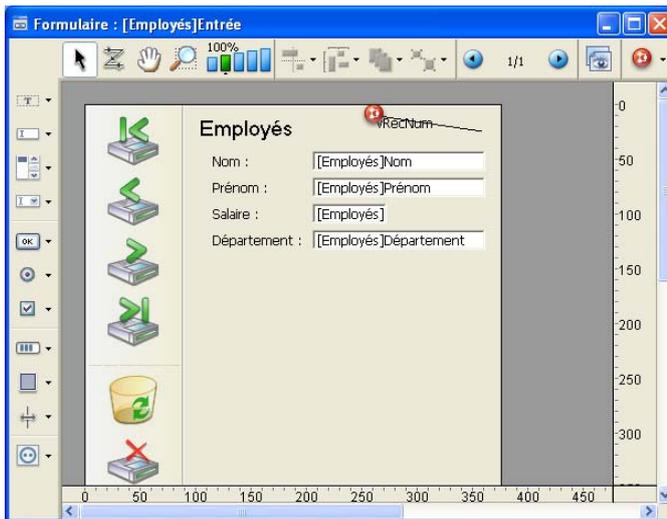
Sur la seconde machine, appuyez sur la touche **Echap**. puis choisissez la commande **Structure** dans le menu **Mode**.

Faites la même chose sur le premier poste.

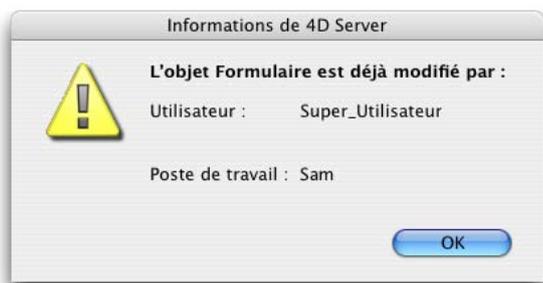
Sur la première machine, choisissez **Explorateur** > **Formulaires** dans le menu **Structure**. La fenêtre de l'Explorateur apparaît. Déployez la table Employés :



Double-cliquez sur le formulaire entrée. L'éditeur de formulaires apparaît, il contient le formulaire entrée :

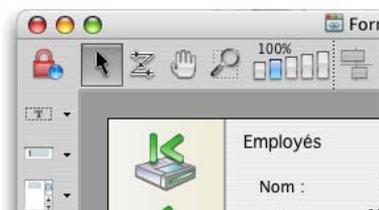


Faites la même chose sur le second poste. Comme le formulaire est déjà en mode modification sur l'autre machine, le mécanisme intégré de verrouillage des objets de 4D Server vous informe :



Malgré tout, vous pouvez ouvrir le formulaire sur la seconde machine, en mode visualisation seulement. Vous pouvez sélectionner et copier des objets vers d'autres formulaires, mais vous ne pouvez pas modifier le formulaire lui-même.

Notez l'icône représentant un cadenas dans l'angle supérieur gauche du formulaire :



Elle signifie que vous ne pouvez pas modifier le formulaire.

Sur le premier poste, sélectionnez le libellé "Nom" à gauche du champ [Employés]Nom. Sélectionnez ensuite le menu hiérarchique **Objet>Couleur**. Affectez, par exemple, la couleur rouge à l'objet :



Sélectionnez ensuite **Sauvegarder le formulaire: [Employés]Entrée** dans le menu **Fichier**.

Sur la seconde machine, refermez puis ouvrez à nouveau le formulaire afin de le charger ; la modification de couleur que vous venez d'apporter au libellé apparaît alors :



The image shows a form titled "Employés" with a red error icon and the text "VRecNum" next to it. Below the title, there are two input fields: "Nom : [Employés]Nom" and "Prénom : [Employés]Prénom".

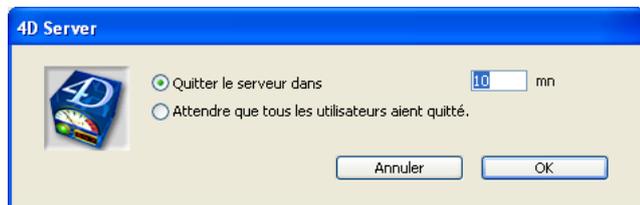
4D Server vous permet de développer simultanément une base à plusieurs !

Arrêt du serveur

En plus de l'information des utilisateurs 4D Client lors d'accès simultanés aux mêmes enregistrements ou objets, 4D Server gère en interne l'information des utilisateurs à travers le réseau lorsqu'il quitte.

Tout en maintenant connectés vos deux clients, choisissez, sur le poste serveur, la commande **Quitter** dans le menu **Fichier** (Windows) ou dans le menu **4D Server** (MacOS).

La boîte de dialogue de fermeture de la base apparaît :



Cliquez sur le bouton **OK**.

Presque instantanément, les deux postes clients sont informés que le serveur va bientôt quitter. Si, par exemple, un client était en train d'ajouter un enregistrement, l'utilisateur disposerait de suffisamment de temps pour terminer et valider la saisie.



Cette boîte de dialogue d'alerte est répétée régulièrement sur chaque poste client.

Note : Vous pouvez également choisir de quitter l'application serveur en utilisant l'option "Attendre que tous les utilisateurs soient déconnectés".

Pendant que le serveur poursuit le processus de déconnexion, quittez 4D Client sur les deux machines.

Et maintenant ?

Après ces neuf minutes intenses et pendant que le serveur est en train de s'arrêter, vous pouvez faire une pause.

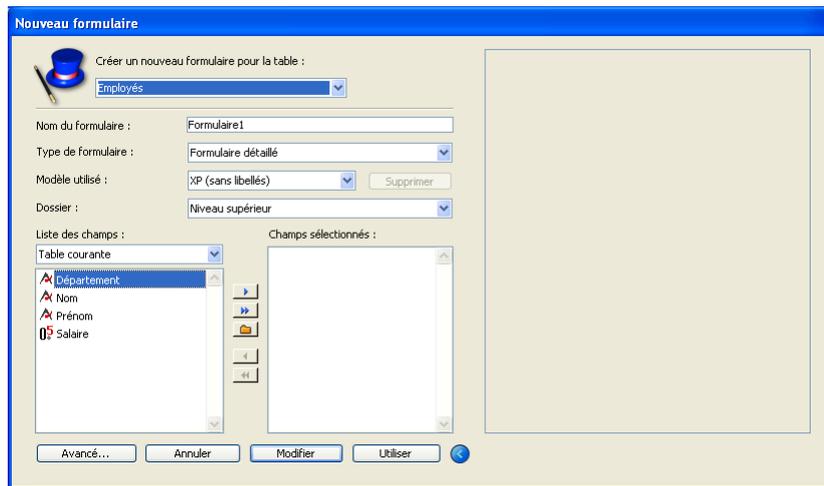
Sachez aussi que, sans le savoir, vous n'avez pas développé seulement deux applications, mais trois !

Pour savoir pourquoi, rendez-vous à la section 4D Server est un serveur Web. Vous en aurez approximativement pour une minute.

Redémarrez la base serveur. Pour cela, lancez 4D Server et ouvrez la base Employés que vous avez créée.

Connectez-vous à la base avec 4D Client. Vous allez créer rapidement un formulaire adapté à la publication Web.

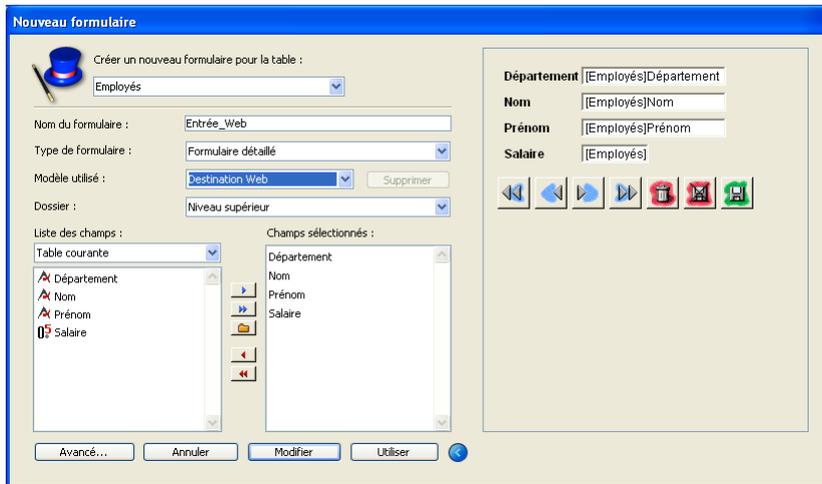
En mode Structure, choisissez la commande **Nouveau>Formulaire...** dans le menu **Fichier**. L'assistant de création de formulaires apparaît.



Effectuez les paramètres suivants :

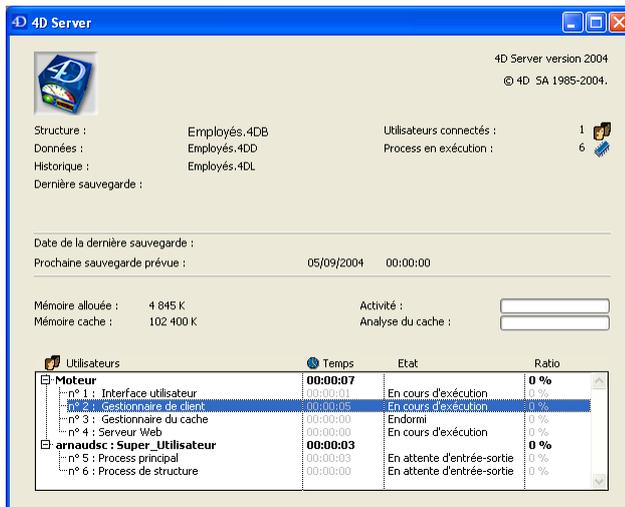
- Nom du formulaire : “Entrée Web”
- Type de formulaire : Formulaire détaillé
- Modèle utilisé : Destination Web
- Liste des champs : cliquez sur le bouton “Ajouter tous les champs”.

L'assistant doit avoir l'apparence suivante :

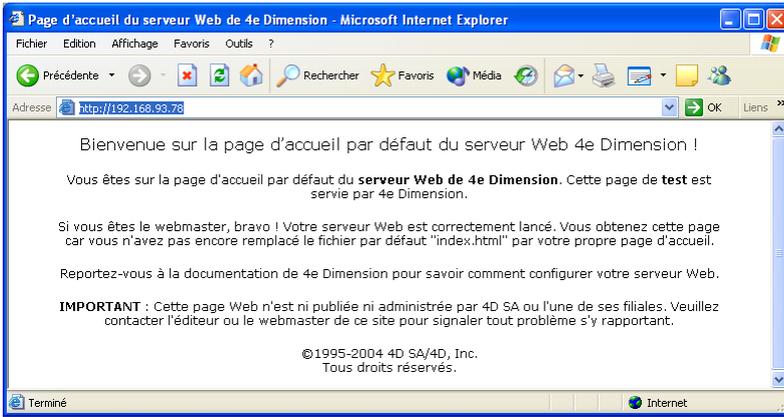


Cliquez sur le bouton **Utiliser**.

Sur le poste serveur, il y a alors un utilisateur connecté :



Pour vérifier que le serveur Web fonctionne, lancez un navigateur à partir d'une autre machine. Dans la zone "Adresse" du navigateur, saisissez l'adresse IP de la machine serveur (généralement du type 123.45.67.89) puis validez. Vous devez obtenir la page d'accueil par défaut du serveur Web :

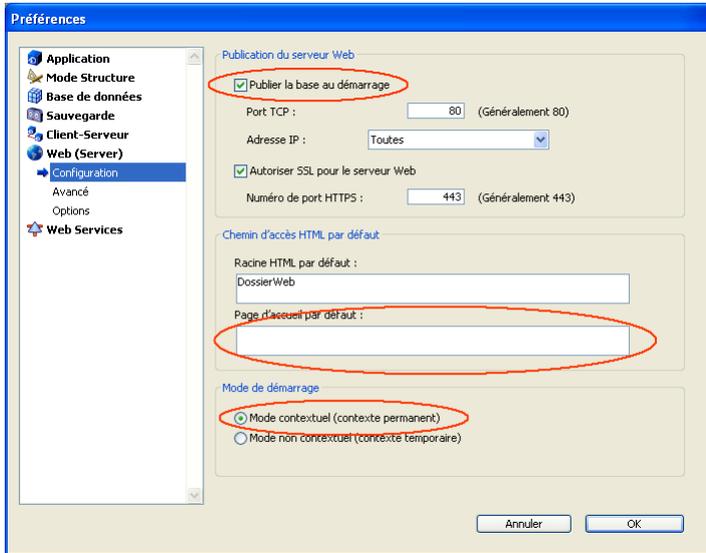


Vous allez maintenant configurer le serveur Web de manière à qu'il démarre en mode contextuel. Dans ce mode, les pages Web sont issues de la conversion dynamique des formulaires et des barres de menus 4D. Le mode contextuel est particulièrement adapté à la publication rapide d'une base existante (pour plus d'informations sur ce mode, reportez-vous à la section Utiliser le mode contextuel dans le manuel Langage de 4D).

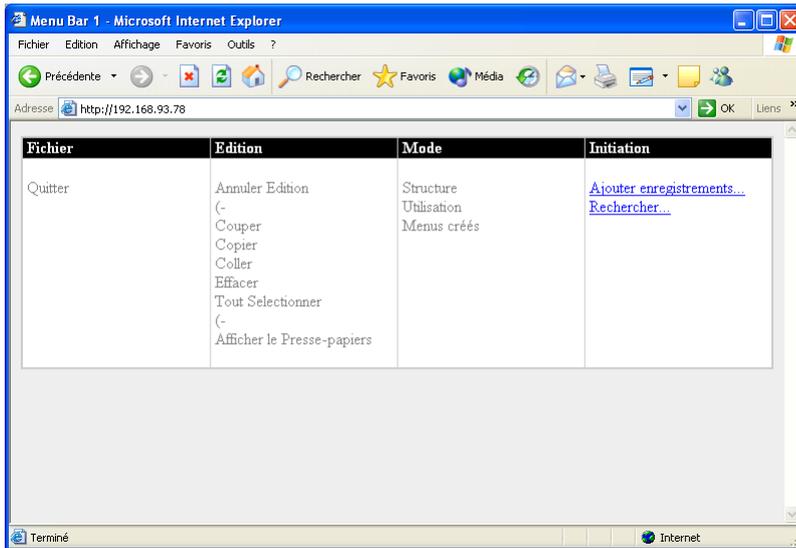
Sur le poste serveur, choisissez **Préférences...** dans le menu **Edition**, affichez la page "Web (Server)/Configuration" et effectuez les modifications suivantes :

- vérifiez que la base est publiée au démarrage,
- supprimez la page d'accueil par défaut (afin d'utiliser la barre de menus comme page d'accueil),
- cochez l'option "Mode contextuel".

La page doit apparaître ainsi :

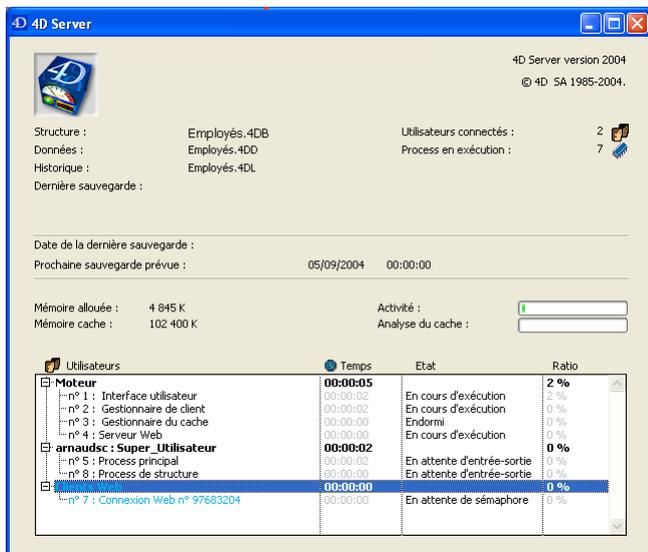


Retournez sur la machine du navigateur Web et actualisez la connexion au serveur. Vous obtenez maintenant la version Web de votre barre de menus personnalisée (il sera peut-être nécessaire de vider le cache du navigateur) :

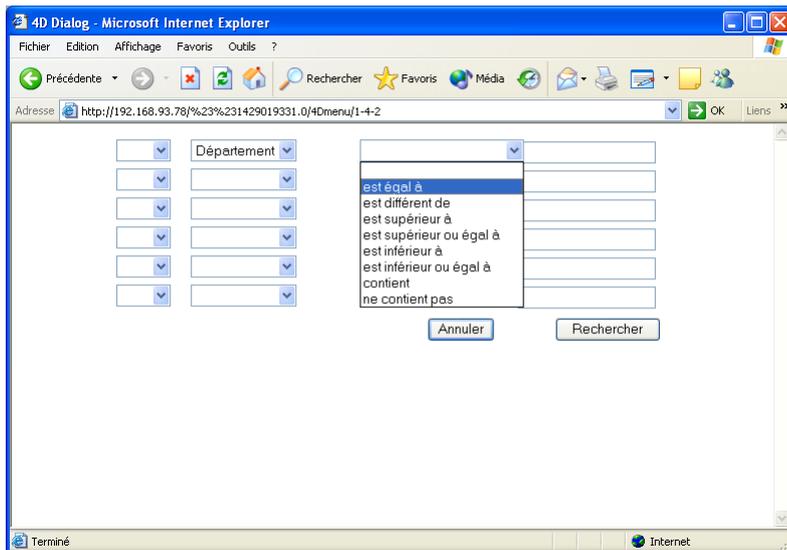


Notez que vous n'avez pas eu besoin d'écrire une seule ligne de code HTML ni de créer un module CGI pour arriver à ce résultat : 4D Server est aussi un serveur Web !

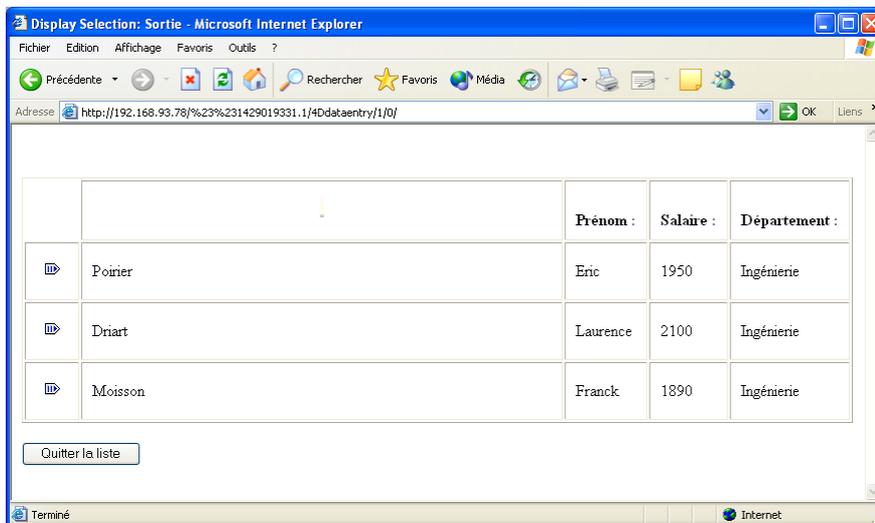
Sur le poste serveur, la connexion Web contextuelle apparaît dans la fenêtre des process :



Dans le navigateur Web, cliquez sur le lien **Rechercher**. 4D Server a traduit à la volée son éditeur de recherches standard en une page HTML. Définissez la recherche "Département est égal à Ingénierie" :

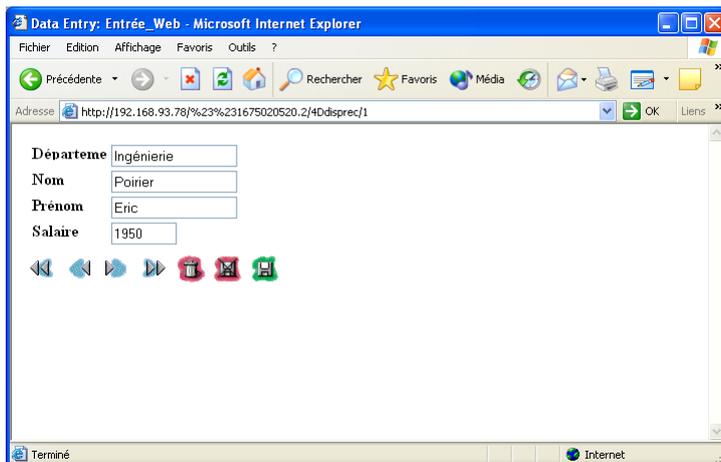


Vous obtenez le résultat suivant :

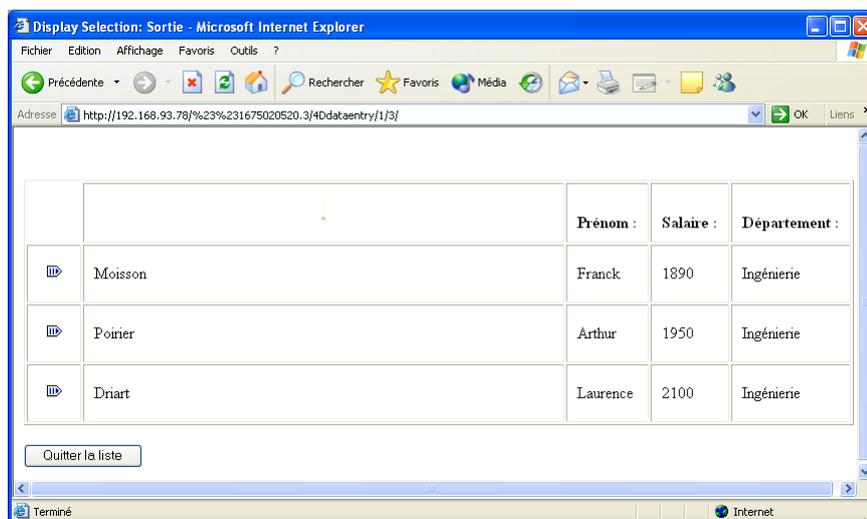


4D Server traduit dynamiquement pour vous les données 4D en données Web et affiche la liste des enregistrements.

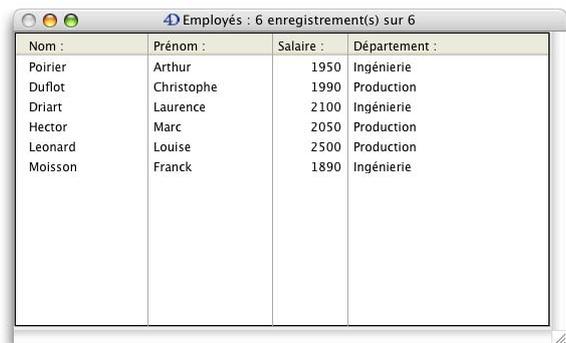
En utilisant l'icône de la colonne de gauche, ouvrez l'enregistrement "Poirier, Eric". Le formulaire entrée que vous utilisez avec 4D Client apparaît en tant que page Web (traduit de manière transparente par 4D Server) :



Changez le prénom en Arthur et validez la saisie. De retour à la liste d'enregistrements, vous constatez que la modification est effectivement reportée dans la liste des enregistrements :



Enfin, si vous vous placez en mode Utilisation sur le poste 4D Client, vous constatez que la modification y est également reportée :



Conclusion

A travers cette initiation (qui a pu prendre en réalité plus de 10 minutes, en fonction de divers facteurs, comme la durée de votre pause), vous avez pu découvrir la simplicité et la facilité d'utilisation de 4D Server :

- Vous avez créé une nouvelle base
- Vous avez créé une table et laissé 4D Server construire les formulaires pour vous
- Vous avez ajouté et manipulé des enregistrements

- Vous avez personnalisé votre application avec votre propre barre de menus
- Vous avez utilisé la base serveur simultanément sous Windows et MacOS
- Vous avez quitté puis relancé le serveur
- Vous avez utilisé la base serveur simultanément avec 4D Client et via le Web !

En définitive, vous avez créé trois applications personnalisées (Windows, Macintosh et Web) alors que vous n'avez en fait effectué qu'un seul développement. De plus, si vous souhaitez utiliser la base en mode monoposte, vous pouvez l'ouvrir directement avec 4e Dimension ou 4D Runtime.

Pour en savoir plus à propos de 4D Server, reportez-vous aux sections d'introduction de ce manuel, ainsi qu'aux autres sections qui détaillent le fonctionnement de 4D Server.

Pour une information complète sur l'environnement 4D, référez-vous aux manuels suivants :

- Mode Structure de 4e Dimension pour savoir comment construire des bases 4D.
- Mode Utilisation de 4e Dimension pour savoir comment utiliser les éditeurs intégrés de 4D tels que l'éditeur de recherches ou de graphes, que vous avez rencontrés dans ce manuel.
- Langage de 4e Dimension pour connaître en détail les commandes du langage de 4D. Si, par exemple, vous souhaitez mieux connaître les capacités de 4D Server en matière de Web, lisez la section Présentation du serveur Web dans le manuel Langage de 4D.

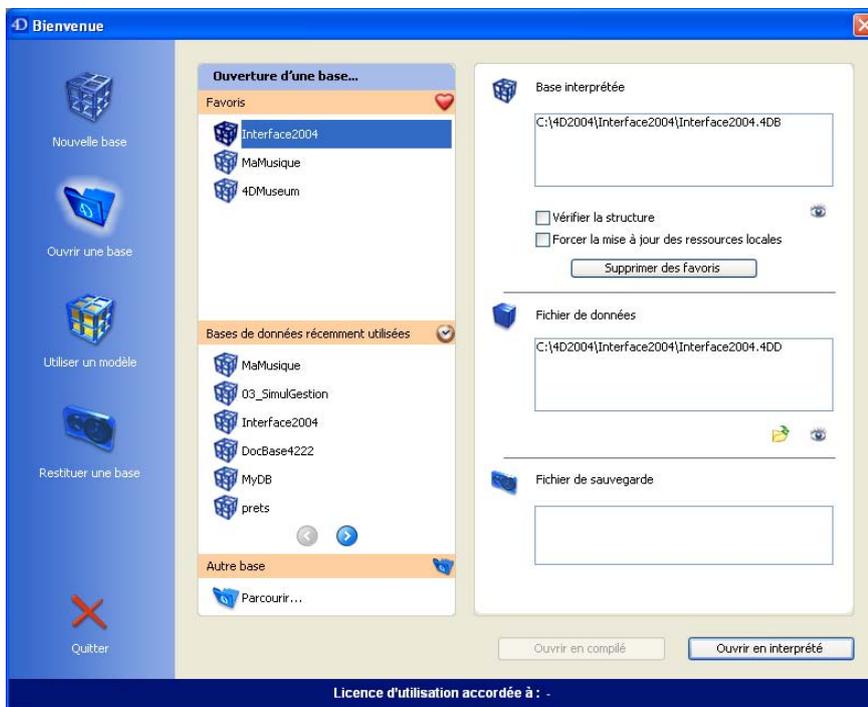
3

Utilisation de 4D Server

Pour créer une nouvelle base ou ouvrir une base existante, vous devez lancer 4D Server en double-cliquant sur l'icône de l'application 4D Server.



La boîte de dialogue de bienvenue apparaît, vous permettant d'ouvrir une base existante, d'en créer une nouvelle (vierge ou basée sur un modèle) ou de restituer une base archivée.



Le fonctionnement de cette boîte de dialogue est identique à celui de la boîte de dialogue de bienvenue de 4e Dimension version monoposte. Pour plus d'informations, reportez-vous au manuel *Mode Structure* de la documentation de 4e Dimension.

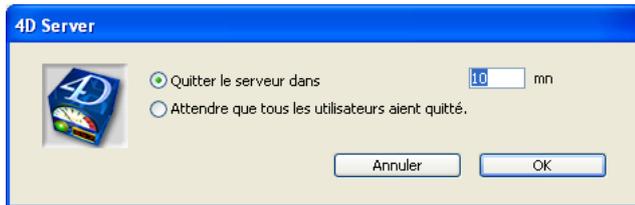
4D Server propose toutefois l'option supplémentaire **Forcer la mise à jour des ressources locales** dans la zone "Base interprétée". Cette option provoque l'envoi systématique des ressources locales à chaque poste client au moment de sa connexion. Les ressources locales sont les informations structurelles relatives à la base stockées sur chaque poste client. La mise à jour des ressources locales est en principe automatique sur les postes clients lorsque la structure de la base a été modifiée depuis la dernière connexion. Dans la plupart des cas, cette option est inutile. Toutefois, dans certains cas spécifiques, il peut être nécessaire de forcer la mise à jour.

Cette option peut également être définie au niveau de chaque poste client, elle s'applique dans ce cas à uniquement au client concerné. Pour plus d'informations, reportez-vous à la section Connexion à une base 4D Server.

Pour quitter l'application 4D Server, procédez ainsi :

1. Sélectionnez la commande Quitter dans le menu Fichier de 4D Server (Windows) ou dans le menu 4D Server (Mac OS).

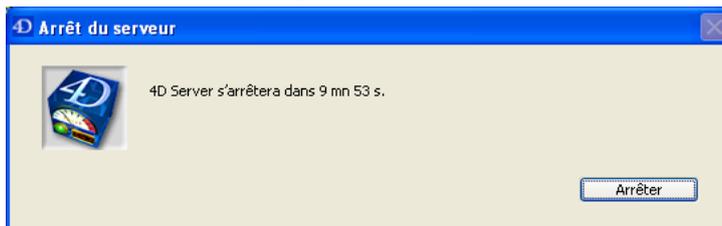
La fenêtre suivante s'affiche sur le poste serveur :



2. Saisissez le nombre de minutes imparties avant que le serveur ne quitte ou sélectionnez l'option "Attendre que tous les utilisateurs aient quitté" puis cliquez sur OK.

A partir de cet instant, plus aucun nouveau client ne peut se connecter au serveur.

- Si vous avez choisi l'option "Quitter le serveur dans XX mn", la fenêtre suivante apparaît :



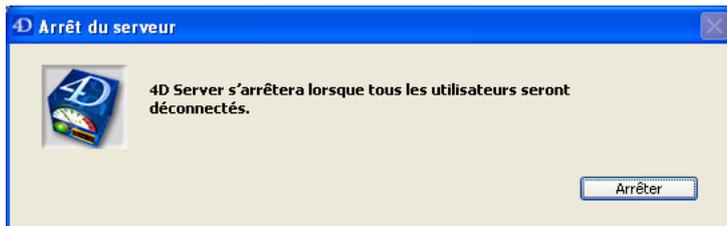
Simultanément, le serveur avertit les postes clients du temps qu'il leur reste avant la déconnexion :



Fenêtre de fermeture s'affichant sur les postes clients

Cette boîte de dialogue est répétée ou mise à jour sur chaque poste client toutes les 20 secondes environ, afin de l'inciter à quitter. A l'issue du délai, le serveur quitte même si des clients sont encore connectés.

- Si vous avez choisi l'option "Attendre que tous les utilisateurs aient quitté", la fenêtre suivante apparaît :



Aucune boîte de dialogue n'apparaît sur les postes clients. Le serveur quitte dès que tous les clients se sont déconnectés.

Notes :

- Dans les deux cas, si aucun client n'est connecté au serveur au moment où la fenêtre de fermeture est validée, 4D Server quitte immédiatement.
- Pour forcer 4D Server à quitter sans délai, passez 0 mn comme délai d'attente.
- Si vous cliquez sur le bouton **Arrêter** dans la fenêtre "Arrêt du serveur", le processus d'arrêt du serveur est annulé.

La fenêtre de 4D Server contient des informations vous permettant de gérer divers aspects de votre système de base de données, à des fins d'administration :

The screenshot shows the 4D Server administration window. At the top left is the 4D logo. The title bar reads '4D Server'. In the top right corner, it says '4D Server version 2004' and '© 4D SA 1985-2004'. Below this, there are statistics: 'Structure : Employés.4DB', 'Données : Employés.4DD', 'Historique : Employés.4DL', 'Dernière sauvegarde :', 'Utilisateurs connectés : 2', and 'Process en exécution : 7'. A section for backups shows 'Date de la dernière sauvegarde :', 'Prochaine sauvegarde prévue : 05/09/2004 00:00:00', 'Mémoire allouée : 4 845 K', 'Mémoire cache : 102 400 K', 'Activité :', and 'Analyse du cache :'. The bottom part of the window is a table titled 'Utilisateurs' with columns for 'Temps' and 'Etat'. The table lists several processes under 'Moteur' and 'arnaudsc : Super_Utilisateur', with their respective execution times and states.

Utilisateurs	Temps	Etat	Ratio
Moteur	00:00:05		2 %
n° 1 : Interface utilisateur	00:00:02	En cours d'exécution	2 %
n° 2 : Gestionnaire de client	00:00:02	En cours d'exécution	0 %
n° 3 : Gestionnaire du cache	00:00:00	Endormi	0 %
n° 4 : Serveur Web	00:00:00	En cours d'exécution	0 %
arnaudsc : Super_Utilisateur	00:00:02		0 %
n° 5 : Process principal	00:00:02	En attente d'entrée-sortie	0 %
n° 8 : Process de structure	00:00:00	En attente d'entrée-sortie	0 %
arnaudsc : Web	00:00:00		0 %
n° 7 : Connexion Web n° 97683204	00:00:00	En attente de sémaphore	0 %

La fenêtre du serveur se divise en deux parties :

- La partie supérieure affiche des informations relatives aux fichiers utilisés par le serveur, aux sauvegardes et à la mémoire
- La partie inférieure affiche des informations sur les clients connectés au serveur et les process en cours d'exécution.

Partie Informations serveur

La partie **Informations serveur** fournit des informations générales sur le serveur :

4D Server version 2004
© 4D SA 1985-2004.

Structure :	Employés.4DB	Utilisateurs connectés :	2
Données :	Employés.4DD	Process en exécution :	4
Historique :	Employés.4DL		
Dernière sauvegarde :	C:\4D2004\Bases\Employés\Employés[0001].4BK		

Date de la dernière sauvegarde :	02/09/2004	15:00:00
Prochaine sauvegarde prévue :	05/09/2004	00:00:00

Mémoire allouée :	4 845 K	Activité :	
Mémoire cache :	102 400 K	Analyse du cache :	

- **Numéro de version de 4D Server** : Numéro de version de l'application 4D Server en cours d'utilisation.
- **Structure** : Nom du fichier de structure ouvert par 4D Server.
- **Données** : Nom du fichier de données associé au fichier de structure.
- **Historique** : Nom du fichier d'historique exploité par la base pour enregistrer les opérations effectuées. Un fichier d'historique est créé par défaut au moment de la création de la base.
- **Dernière sauvegarde** : Chemin d'accès et nom du dernier fichier de sauvegarde de la base.
- *Zone de progression de la sauvegarde* : Cette zone affiche un thermomètre de progression lorsqu'une sauvegarde est en cours. Le bouton en forme de X permet de stopper manuellement une sauvegarde automatique en cours d'exécution :

Sauvegarde en cours...

Date de la dernière sauvegarde : 17/08/2006 14:28:00

- **Date de la dernière sauvegarde** : Date et heure auxquelles la dernière sauvegarde de la base a été effectuée.
- **Prochaine sauvegarde prévue** : Date et heure de la prochaine sauvegarde périodique automatique. Cette information est issue de la configuration des sauvegardes périodiques définie dans la boîte de dialogue des Préférences de la base.
- **Utilisateurs connectés** : Nombre total d'utilisateurs actuellement connectés à la base.
- **Process en exécution** : Nombre total de process en cours d'exécution. Ce nombre comprend tous les process (moteur, connexions d'utilisateurs, connexions Web et procédures stockées).

- **Mémoire allouée** : Mémoire allouée au moteur de 4D Server. Cette valeur est définie dynamiquement par le Système d'exploitation en fonction de l'utilisation de la base et des ressources disponibles.
- **Mémoire cache** : Mémoire allouée au cache. La taille du cache détermine la quantité de données qui peut être conservée en mémoire afin de réduire les accès disque. La mémoire cache peut être configurée dans la page Base de données/Gestion des données des Préférences de l'application. Pour plus d'informations sur ce point, reportez-vous au manuel Mode Structure de la documentation de 4D.
- **Thermomètre Activité** : Thermomètre indiquant le niveau courant d'activité du serveur. Ce thermomètre indique l'activité totale sur le réseau. Plus il y a de requêtes envoyées au serveur, plus la valeur du thermomètre est élevée.
- **Thermomètre Analyse du cache** : Thermomètre indiquant le niveau d'utilisation du cache. Plus la valeur du thermomètre est élevée, plus le cache est utilisé. Si le thermomètre est toujours à un bas niveau, vous pouvez envisager de réduire sa taille.

Fenêtre Mémoire cache

Le thermomètre "Analyse du cache" peut également être utilisé en tant que bouton. Lorsque vous cliquez dessus, la fenêtre Mémoire cache apparaît :



Cette fenêtre affiche des thermomètres supplémentaires détaillant l'utilisation du cache. Un thermomètre indique la taille de chaque type de données pouvant être stocké dans le cache. Une valeur élevée est considérée comme avantageuse car cela signifie que le type de données nécessitera moins d'accès disques.

Les enregistrements, les adresses des enregistrements et les adresses des pages d'index sont chargés progressivement dans le cache lorsque vous travaillez avec les enregistrements. Les pages d'index et les tables d'allocations sont chargées progressivement dans le cache lorsque vous modifiez les données en ajoutant, modifiant ou supprimant des enregistrements.

Partie Informations process

La partie **Information process** liste les utilisateurs connectés et les process en cours d'exécution :

Utilisateurs	Temps	Etat	Ratio
Moteur	00:00:42		1 %
n° 1 : Interface utilisateur	00:00:38	En cours d'exécution	1 %
n° 2 : Gestionnaire de client	00:00:02	En cours d'exécution	0 %
n° 3 : Gestionnaire du cache	00:00:00	Endormi	0 %
n° 4 : Serveur Web	00:00:00	En cours d'exécution	0 %
arnaudsc : Super Utilisateur (192.168.93.9)	00:00:00		0 %
n° 5 : Process principal	00:00:00	En attente d'entrée-sortie	0 %
n° 6 : Process de structure	00:00:00	En attente d'entrée-sortie	0 %
Clients Web	00:00:00		0 %
n° 9 : Connexion Web n° 1348412964	00:00:00	En attente de sémaphore	0 %
Procédures stockées	00:00:15		94 %
n° 7 : Méthode 1	00:00:15	En cours d'exécution	94 %

Tous les process clients, les **process de connexion** (à l'exception des process locaux des postes clients), sont inclus dans la liste. Vous pouvez déployer et contracter la liste des process du moteur ou d'un utilisateur particulier en cliquant sur le signe + (Windows) ou la flèche (Macintosh) situé(e) à gauche du nom de l'utilisateur ou de l'intitulé Moteur. L'adresse IP de chaque utilisateur connecté à la base via 4D Client est affichée entre parenthèses derrière le nom du poste client (depuis la version 2004.5). Lorsqu'aucun client n'est connecté, seules les informations sur l'activité du moteur de la base sont affichées, ainsi que (éventuellement) les procédures stockées.

Le moteur de base de données de 4D Server est géré par quatre process :

- **Interface utilisateur** : contrôle l'interface utilisateur de la fenêtre du serveur elle-même.
- **Gestionnaire de client** : gère toutes les connexions au serveur et l'allocation de temps de traitement pour chaque client.
- **Gestionnaire du cache** : gère l'écriture des données sur le disque.
- **Serveur Web** : Le rôle du process Serveur Web est uniquement de gérer les tentatives de connexion par le Web. L'exécution du process Serveur Web ne signifie pas qu'une session Web a été ouverte, mais simplement que vous avez autorisé les utilisateurs Web à démarrer des connexions Web.

Pour chaque process, la fenêtre du serveur fournit les informations suivantes :

- Nom du process,
- Temps (en secondes) d'exécution du process depuis sa création,
- Etat courant du process,
- Pourcentage du temps que 4D Server a consacré à ce process (ratio).

Afin de vous permettre de différencier les types de process, la fenêtre des process affiche les informations de la manière suivante :

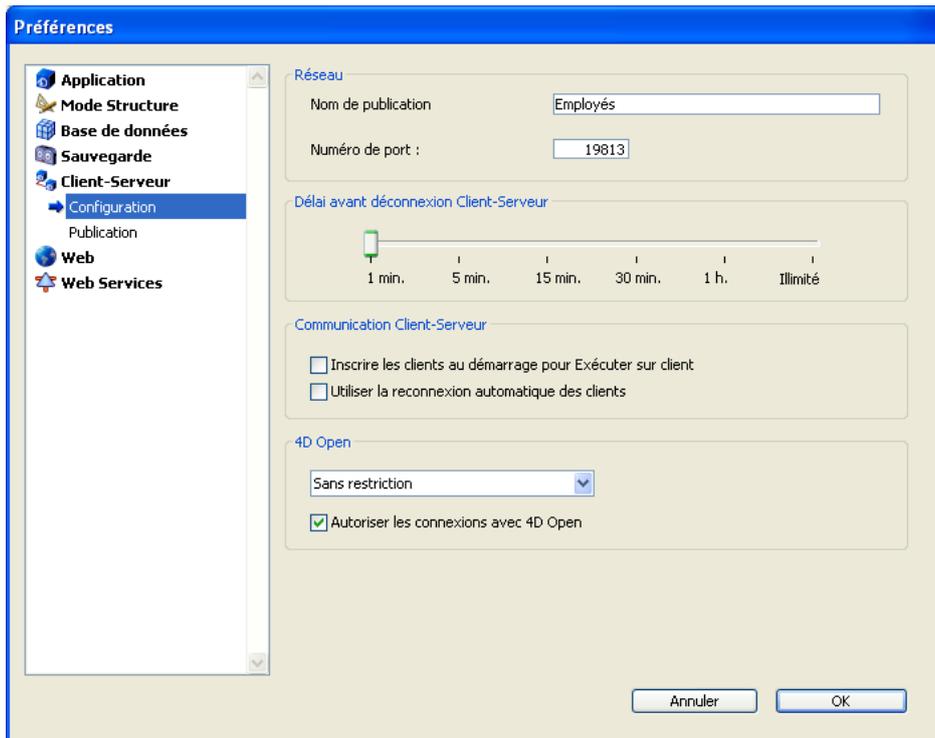
- Process moteur en noir
- Process de connexion utilisateur en noir
- Process de connexion Web en cyan
- Process de procédure stockée en bleu
- Process basés sur des connexions 4D Open en vert.

Note : Les process en train d'être refermés peuvent apparaître pendant un court laps de temps en rouge.

Référence

Introduction aux process, Menu Process.

Vous pouvez définir divers paramètres relatifs au réseau et à la communication client-serveur dans la page “Configuration” du thème **Client-Serveur** des Préférences de l'application (accessibles depuis 4D Client et 4D Server) :



Ces paramètres sont décrits dans cette section.

Nom de publication

Cette option permet de modifier le nom de publication d'une base publiée par 4D Server, c'est-à-dire le nom affiché dans la page de publication dynamique TCP/IP de la boîte de dialogue de connexion (cf. section Connexion à une base 4D Server).

Par défaut, 4D Server utilise le nom du fichier de structure de la base. Vous pouvez saisir tout nom personnalisé.

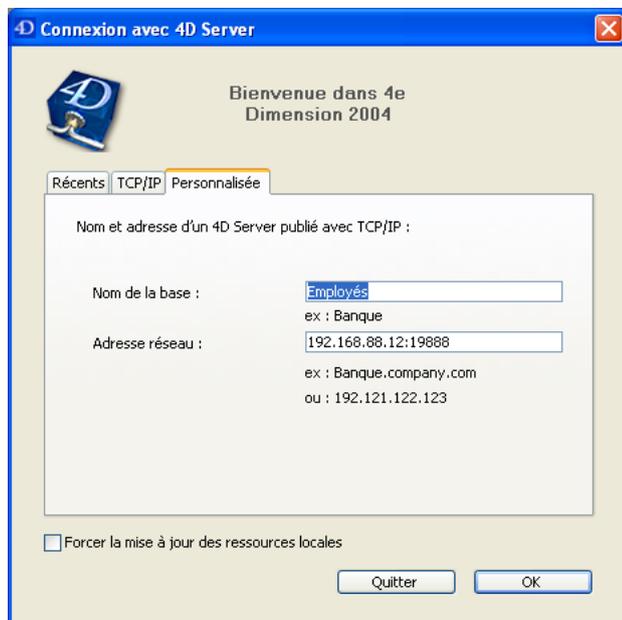
Note : Ce paramètre n'est pas pris en compte dans le cadre des applications client-serveur personnalisées. En principe, l'application cliente se connecte directement à l'application serveur, sans passer par la boîte de dialogue de connexion. Toutefois, en cas d'erreur, cette boîte de dialogue apparaît ; dans ce cas, le nom de publication de l'application serveur est le nom de la base compilée.

Numéro de port

Cette option permet de modifier le numéro de port TCP sur lequel 4D Server publie la base de données. Cette information est stockée dans la structure de la base et sur chaque poste client. Par défaut, le numéro de port TCP utilisé par 4D Server et 4D Client est le 19813.

La personnalisation de cette valeur est nécessaire lorsque vous souhaitez utiliser plusieurs applications 4D sur la même machine avec le protocole TCP ; dans ce cas, vous devez spécifier un numéro de port différent pour chaque application.

Lorsque vous modifiez cette valeur depuis 4D Server ou 4D Client, elle est automatiquement répercutée sur tous les postes 4D Client connectés à la base. Pour mettre à jour les autres postes clients non connectés, il suffira, lors de leur connexion suivante, de saisir le nouveau numéro de port (précédé de deux-points) derrière l'adresse IP du poste serveur dans la page Personnalisée de la boîte de dialogue de connexion. Par exemple, si le nouveau numéro de port est le 19888 :



Note : Seules les bases publiées sur le même port que celui défini dans 4D Client sont visibles dans la page de publication dynamique TCP/IP.

Délai avant déconnexion Client-Serveur

Ce thermomètre permet de définir le timeout (période d'inactivité au-delà de laquelle la connexion est fermée) entre 4D Server et les postes clients qui s'y connectent.

L'option Illimité élimine le timeout. Lorsque cette option est sélectionnée, le contrôle d'inactivité du client est désactivé.

Lorsqu'un délai est sélectionné, le serveur mettra un terme à la connexion d'un client s'il ne reçoit pas de requête de ce dernier dans l'intervalle de temps spécifié.

Inscrire les clients au démarrage pour Exécuter sur client

Lorsque cette option est cochée, tous les postes 4D Client se connectant à la base peuvent exécuter des méthodes à distance. Ce mécanisme est détaillé dans la section Inscrire les clients.

Utiliser la reconnexion automatique des clients

Cette fonction bas niveau permet, dans certaines configurations spécifiques, de reconnecter automatiquement les postes clients en cas de déconnexion inopinée.

Lorsque l'option (globale à tous les clients) est cochée, la fonction de reconnexion est activée automatiquement en cas de perte de la connexion d'un client avec le serveur. Une seule tentative est effectuée : si la reconnexion échoue, l'erreur -10002 est retournée. En cas de succès, une boîte de dialogue d'alerte est affichée et, sous Windows, une icône de notification s'inscrit dans la barre des tâches.

Pour que le mécanisme de reconnexion automatique puisse être mis en oeuvre, le délai avant déconnexion (timeout) côté serveur doit être supérieur à 1 minute.

4D Open

4D Open est l'API (Application Programming Interface) qui permet à des applications non-4D Client de se connecter à 4D Server.

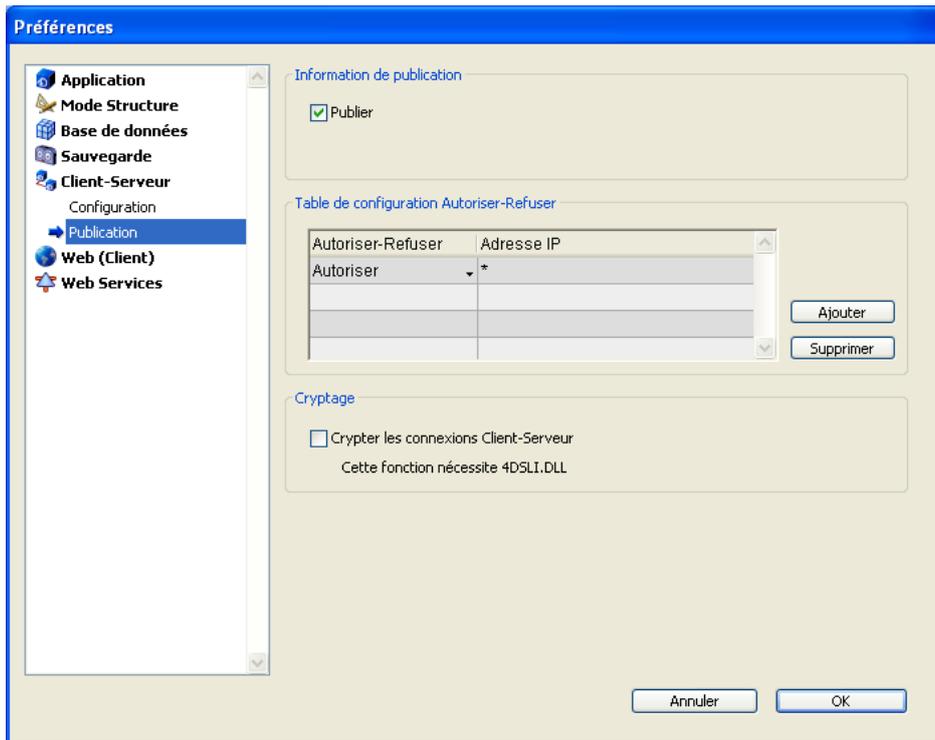
- **Autoriser les connexions avec 4D Open** : cette option, lorsqu'elle est cochée, donne au groupe spécifié par le menu la possibilité de se connecter à 4D Server à partir d'une application 4D Open.
- **Menu d'accès par 4D Open** : ce menu permet de désigner le groupe d'utilisateurs autorisé à se connecter à la base 4D Server via 4D Open, lorsque l'option "Autoriser les connexions avec 4D Open" est cochée.

Note : Pour des raisons de sécurité, les paramètres d'accès 4D Open ne sont pas modifiables sur le poste serveur.

Référence

Inscrire les clients, Préférences de publication.

Vous pouvez définir les paramètres relatifs à la sécurité et la publication des bases par 4D Server dans la page "Publication" du thème **Client-Serveur** des Préférences de l'application (accessibles depuis 4D Client et 4D Server) :



Ces paramètres sont décrits dans cette section.

Publier

Cette option permet d'indiquer si la base 4D Server doit apparaître ou non dans la liste des bases publiées dans la boîte de dialogue de connexion.

- lorsque l'option est cochée (par défaut), la base est rendue publique, elle apparaît dans la liste des bases publiées (page **TCP/IP**).
- lorsque l'option est désélectionnée, la base n'est pas rendue publique, elle n'apparaît pas dans la liste des bases publiées. Pour se connecter, les utilisateurs doivent saisir manuellement l'adresse de la base dans la page **Personnalisée** de la boîte de dialogue de connexion.

Table de configuration Accepter/Refuser

Cette table vous permet de définir des règles de contrôle d'accès à la base en fonction de l'adresse IP des postes 4D Client. Cette option permet de renforcer la sécurité par exemple pour des applications stratégiques.

Note : Cette table de configuration ne contrôle pas les connexions Web.

Le fonctionnement de la table de configuration est le suivant :

- La colonne "Autoriser-Refuser" permet de sélectionner le type de règle à appliquer (Autoriser ou Refuser) à l'aide d'un pop up menu. Pour ajouter une règle d'adresses, cliquez sur le bouton Ajouter. Une nouvelle ligne apparaît dans la table. Le bouton Supprimer permet de supprimer la ligne courante.
- La colonne "Adresse IP" permet de désigner la ou les adresse(s) IP concernées par la règle. Pour spécifier une adresse, cliquez dans la colonne et saisissez l'adresse sous la forme 123.45.67.89.

Vous pouvez utiliser le caractère * (étoile) pour spécifier des adresses du type "commence par". Par exemple, 192.168.* indique toutes les adresses débutant par 192.168.

- L'application des règles s'effectue dans l'ordre d'affichage de la table. Si deux règles sont contradictoires, la priorité sera accordée à la règle située le plus haut dans le tableau.

Vous pouvez réordonner les lignes en modifiant le tri courant (cliquez sur un en-tête de colonne pour alterner le sens de tri). Vous pouvez également déplacer des lignes par glisser-déposer.

- Pour des raisons de sécurité, seules les adresses correspondant à une règle d'autorisation explicite pourront se connecter.

En particulier, si la table contient uniquement une ou plusieurs règle(s) de type Refuser, toutes les adresses seront refusées car aucune ne satisfera à au moins une règle. Si vous souhaitez refuser certaines adresses et autoriser toutes les autres, ajoutez une règle Autoriser * à la fin de la table. Par exemple :

- Refuser 192.168.* (*refuser toutes adresses débutant par 192.168*)
- Autoriser * (*et autoriser les autres*)

Par défaut, aucune restriction de connexion n'est appliquée par 4D Server : la première ligne de la table contient le libellé Autoriser et le caractère * (toute adresse).

Crypter les connexions Client-Serveur

Cette option permet d'activer le mode SSL (mode sécurisé) pour la communication entre le poste serveur et les postes 4D Client. Cette option est détaillée dans la section Crypter les connexions client/serveur.

Référence

Crypter les connexions client/serveur, Préférences de configuration.

Vous pouvez configurer vos connexions client/serveur de manière à ce que 4D Server et les postes 4D Client communiquent en mode sécurisé.

La communication client/serveur sécurisée s'appuie sur le protocole SSL (*Secured Socket Layer*).

Qu'est-ce que le protocole SSL dans le cadre des connexions client/serveur ?

Le protocole SSL a pour but de sécuriser les communications entre deux applications — généralement un serveur Web et un navigateur. Il comporte diverses fonctions permettant d'identifier les intervenants, de crypter les données ou encore de vérifier l'intégrité des données échangées. Pour une description détaillée du protocole SSL, reportez-vous, dans le manuel *Langage* de 4D, à la section Utiliser le protocole SSL.

Avec 4D Server et 4D Client, les mécanismes de génération de clés et d'authentification sont "transparents", ils sont gérés par 4D Server et ne nécessitent pas d'intervention de l'utilisateur.

Le cryptage des connexions client/serveur permet donc de renforcer la sécurité de votre application 4D Server, toutefois ce mode de fonctionnement ralentit les échanges.

Configuration

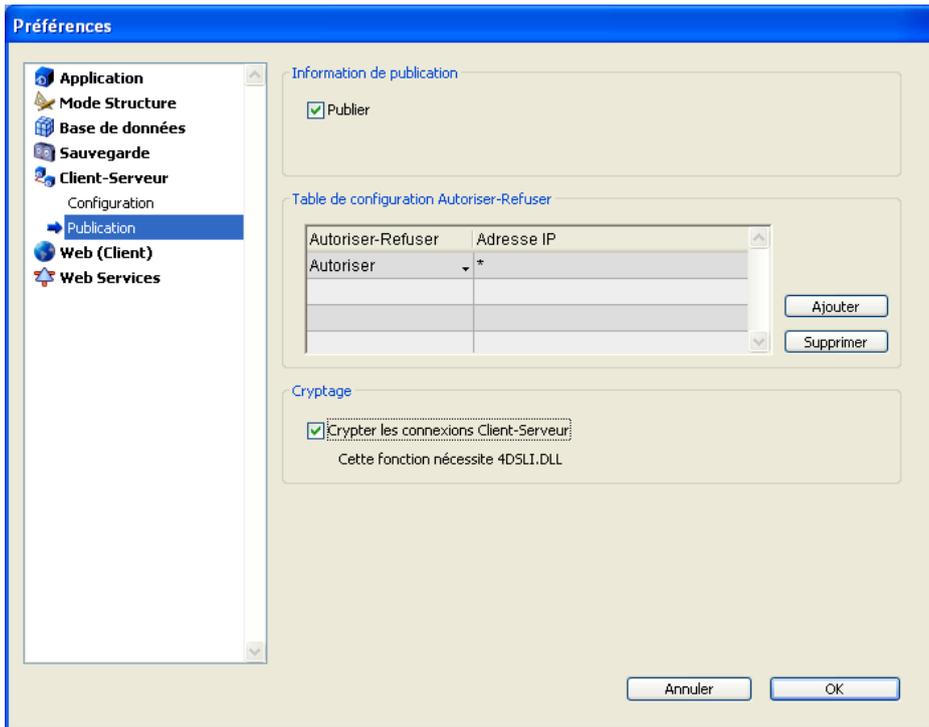
Au niveau de l'architecture réseau, le protocole SSL s'insère entre la couche TCP/IP (bas niveau) et le protocole de haut niveau.

Pour pouvoir utiliser SSL dans une configuration client/serveur, assurez-vous de disposer, sur le poste serveur et sur chaque poste client, du fichier 4DSL1.DLL. Ce fichier constitue l'interface de la couche sécurisée (Secured Layer Interface) dédiée à la gestion du SSL. Il doit se trouver dans le dossier [4D Extensions] de l'application/du progiciel 4D Server et des applications/progiciels 4D Client.



Ce fichier est installé par défaut.

Vous devez en revanche activer les connexions en mode sécurisé.
Pour cela, ouvrez la page “Client-Serveur/Publication” de la boîte de dialogue des Préférences de l'application et cochez l'option **Crypter les connexions Client-Serveur** dans la zone “Cryptage” :

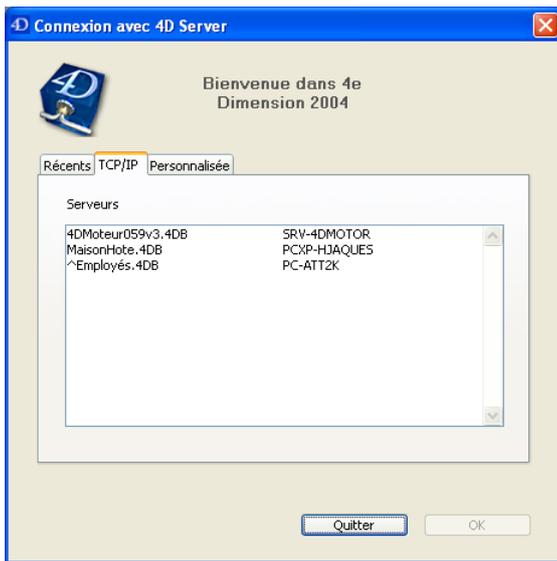


Par défaut, l'option n'est pas cochée.
Vous devez ensuite quitter puis relancer l'application 4D Server pour que ce paramètre soit pris en compte.

Les postes 4D Client se connectent dès lors en mode sécurisé.

Connexion en mode sécurisé

Le symbole “accent circonflexe” (^) apparaît devant le nom des bases publiées en mode SSL dans la page de connexion TCP/IP à 4D Server. Les bases publiées en mode sécurisé apparaissent donc à la fin de la liste :



Note : Lorsque le nom d'une base n'apparaît pas dynamiquement dans la page de connexion TCP/IP, l'utilisateur peut le saisir manuellement dans la page **Personnalisée** (voir sections Connexion à une base 4D Server et Préférences de publication). Dans ce cas, il faut impérativement faire débiter le nom par un ^ si la base est publiée en mode sécurisé, sinon la connexion sera refusée.

Référence

Utiliser le protocole SSL.

Il est possible d'utiliser un dossier spécifique pour transférer tout élément personnalisé du serveur vers les postes clients (fichier de ressources, documents textes, fichiers de préférences xml, etc.).

Ce dossier, nommé **Extras**, doit simplement être placé à côté du fichier de structure. 4D Server gère automatiquement les modifications apportées à ce dossier et ne transfère que ce qui est nécessaire. En outre, le contenu du dossier est compressé afin d'optimiser les temps de copie par le réseau.

Note : Evitez de transférer des données trop volumineuses, car dans ce cas la durée de la phase d'initialisation de la connexion des postes clients pourrait augmenter sensiblement.

Côté client, le dossier Extras est téléchargé au même emplacement que les autres éléments de la structure, c'est-à-dire :

Système	Emplacement des dossiers téléchargés sur le poste client
Mac OS	{Disque}:Utilisateurs:Nom de l'utilisateur:Library:Application Support:4D:NomDeLaBase_Adresse:Extras
Windows 2000 et Windows XP	{Disque}: \Documents and Settings \Nom de l'utilisateur \Application Data \4D \NomDeLaBase_Adresse \Extras

La hiérarchie de départ est reconstituée sur chaque poste client.

4D Server propose une solution intégrée permettant de mettre en place un système de sauvegarde des données par miroir logique. Cette solution est basée sur deux commandes : Nouveau fichier historique et INTEGRER FICHIER HISTORIQUE.

Qu'est-ce qu'un miroir logique ?

Le miroir logique est un mode de sauvegarde sophistiqué, principalement destiné aux bases de données critiques ou à forte charge d'exploitation.

Utiliser un miroir logique consiste à exploiter une base sur un premier poste, et à maintenir sur une deuxième machine une copie de cette base, périodiquement mise à jour. Les deux machines communiquent par le réseau, la machine en exploitation transmettant régulièrement à la machine miroir les évolutions de la base par l'intermédiaire du fichier d'historique.

De cette façon, en cas d'un incident sur la base en exploitation, il n'y a qu'à repartir de la base miroir pour reprendre très rapidement l'exploitation sans aucune perte de données. En outre, la base en exploitation n'est jamais "bloquée" par les opérations de sauvegarde.

Pourquoi choisir la sauvegarde par miroir logique ?

L'emploi d'un miroir logique correspond à des besoins spécifiques. La stratégie standard basée sur une sauvegarde périodique et le fichier d'historique constitue dans la plupart des cas une solution simple, fiable et peu coûteuse. La base est sauvegardée régulièrement (toutes les 24 heures en général). Durant la sauvegarde, la base reste accessible en mode lecture seulement. Cette période d'indisponibilité partielle est très courte, et même pour les bases de données importantes (plus de 2 Go) elle ne dépasse pas 5 minutes. Cette opération peut même être programmée en-dehors des périodes d'utilisation de la base.

Cependant, pour certains organismes, comme par exemple les hôpitaux, les bases de données critiques doivent être entièrement opérationnelles 24h/24. La base ne peut pas être en lecture seulement, même durant un laps de temps très court. Dans ce cas, la mise en place d'un miroir logique est une solution adaptée.

Note : La base miroir ne reflète que les modifications apportées aux **données**. Ce mode de sauvegarde n'est donc pas adapté pour des bases en cours de développement, où les fréquentes modifications de structure rendront rapidement le miroir obsolète ou nécessiteront de multiples réactualisations de la structure de la base miroir.

Principes de fonctionnement

La mise en place d'un système de sauvegarde par miroir logique s'appuie sur deux commandes : Nouveau fichier historique et INTEGRER FICHIER HISTORIQUE. Ces commandes sont décrites dans le manuel *Langage* de 4D.

Les principes mis en oeuvre sont les suivants :

- La base est installée sur le poste 4D Server principal (poste en exploitation) et une copie identique de la base est installée sur le poste 4D Server miroir.
- Un test au démarrage de l'application (par exemple la présence d'un fichier spécifique dans le dossier 4D Extensions) permet de distinguer chaque version (en exploitation et en miroir) et donc d'exécuter les opérations appropriées.
- Sur le poste 4D Server en exploitation, le fichier d'historique est "segmenté" à intervalle régulier à l'aide de la commande Nouveau fichier historique. Aucune sauvegarde n'étant effectuée sur le serveur principal, la base est en permanence disponible en lecture-écriture.
- Chaque "segment" de fichier d'historique est envoyé sur le poste miroir, où il est intégré à la base miroir à l'aide de la commande INTEGRER FICHIER HISTORIQUE.

La mise en place de ce système nécessite la programmation de code spécifique, notamment :

- un minuteur sur le serveur principal pour la gestion des cycles d'exécution de la commande Nouveau fichier historique,
- un système de transfert des "segments" de fichier d'historique entre le poste en exploitation et le poste miroir (utilisation de 4D Internet Commands pour un transfert via ftp ou messagerie, Web Services, 4D Open pour 4D...),
- un process sur le poste miroir destiné à superviser l'arrivée de nouveaux "segments" de fichier d'historique et à les intégrer via la commande INTEGRER FICHIER HISTORIQUE,
- un système de communication et de gestion d'erreurs entre le serveur principal et le serveur miroir.

ATTENTION : La sauvegarde par miroir logique est incompatible avec les sauvegardes "standard" sur la base en exploitation car l'emploi simultané de ces deux modes de sauvegarde entraîne la désynchronisation de la base en exploitation et de la base miroir. Par conséquent, vous devez veiller à ce qu'aucune sauvegarde, automatique ou manuelle, ne soit effectuée sur la base en exploitation. En revanche, il est possible de sauvegarder la base miroir (cf. paragraphe suivant).

Sauvegarde de la base miroir

A compter de la version 2004.4, 4D Server permet d'effectuer des sauvegardes de la base sur le poste miroir.

Tous les moyens classiques peuvent être utilisés pour effectuer les sauvegardes sur le poste miroir : sauvegarde manuelle via la commande du menu **Fichier**, sauvegarde périodique définie dans les Préférences ou sauvegarde programmée à l'aide des commandes du langage. Pour éviter tout risque de désynchronisation avec le poste en exploitation, 4D verrouille automatiquement le poste miroir lorsqu'il effectue l'une ou l'autre des deux opérations fondamentales : intégration de l'historique en provenance du poste en exploitation et sauvegarde de la base miroir.

- Lorsqu'une intégration de l'historique est en cours, il n'est pas possible de déclencher une sauvegarde. Si la commande SAUVEGARDER est utilisée, l'erreur 1417 est générée (cf. section Erreurs du gestionnaire de sauvegarde).

- Lorsqu'une sauvegarde est en cours, tous les process sont gelés, il n'est donc pas possible de démarrer une intégration d'historique.

Scénario d'exploitation d'un miroir logique

Le scénario suivant illustre, du point de vue de chaque poste 4D Server, la mise en place d'un système de sauvegarde avec miroir :

Poste en exploitation

Démarrage de l'application, sauvegarde du fichier de données et activation (si nécessaire) du fichier d'historique.
4D crée le fichier MaBase.4DL

On quitte l'application

Copie de tous les fichiers de la base (fichier d'historique inclus) sur le poste miroir

Démarrage de l'application et passage en exploitation.

Décision de mettre à jour le miroir (par exemple, après une certaine durée d'exploitation)

Exécution de la méthode contenant la commande Nouveau fichier historique. Le fichier sauvegardé est nommé MaBase[0001-0000].4DL

Envoi par programmation du fichier MaBase[0001-0000].4DL au poste miroir (utilisation de 4DIC, 4D Open pour 4D, etc.)

La base est en exploitation

Incident sur le poste, la base de données est inutilisable. Décision de passer sur la machine miroir

Copie du fichier d'historique courant MaBase.4DL sur le poste miroir, dans le dossier de réception habituel

Poste miroir

Démarrage de l'application miroir. 4D Server demande le fichier d'historique : sélection du fichier MaBase.4DL transféré depuis le poste en exploitation.

Détection qu'un fichier est en attente d'intégration. Exécution de la méthode contenant la commande INTEGRER FICHER HISTORIQUE pour intégrer le fichier MaBase[0001-0000].4DL. Ce fichier devient le nouveau fichier d'historique courant.

Réparation de la machine...

Détection qu'un fichier est en attente d'intégration. Exécution de la méthode contenant la commande INTEGRER FICHIER HISTORIQUE pour intégrer le fichier MaBase.4DL. Ce fichier devient le nouveau fichier d'historique courant.

La base est en exploitation

La machine est réparée

On quitte la base

Remplacement des fichiers de la base par ceux de la base miroir

Démarrage de l'application. 4D Server demande le fichier d'historique : sélection du fichier MaBase.4DL transféré depuis le poste miroir.

Démarrage de l'application miroir.

La base est en exploitation

Référence

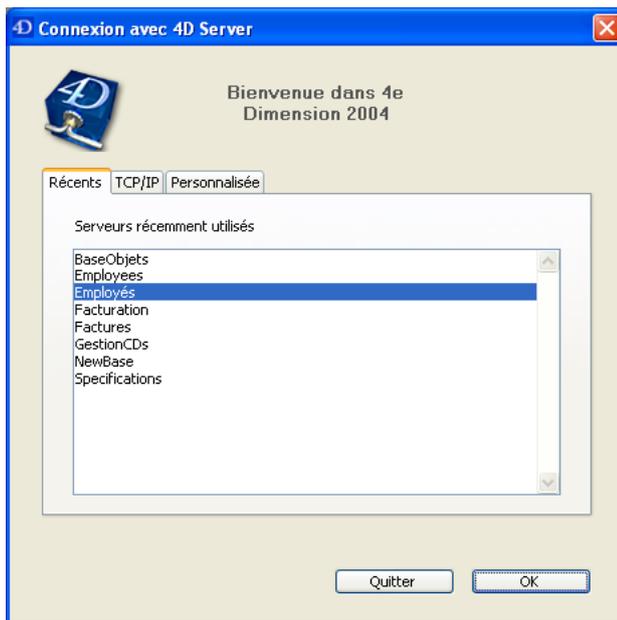
INTEGRER FICHIER HISTORIQUE, Nouveau fichier historique.

4

Utilisation de 4D Client

Lorsque vous lancez 4D Client, une boîte de dialogue de connexion comportant trois pages, accessibles via des onglets, s'affiche.

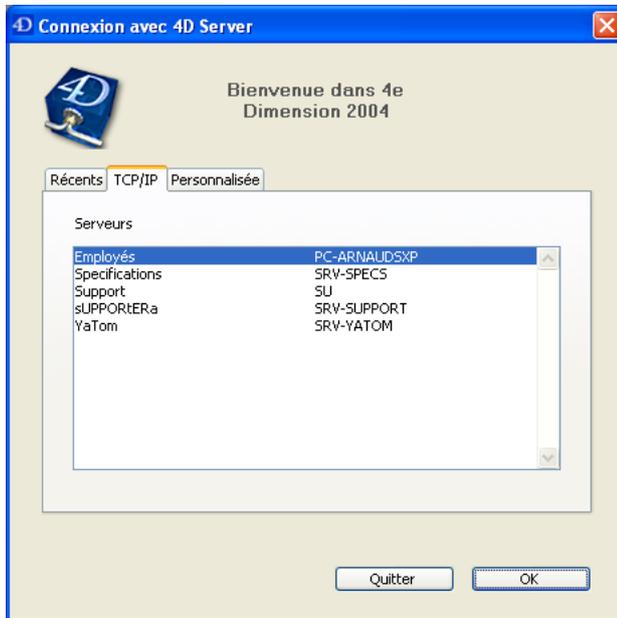
Page "Récents"



La page **Récents** mémorise la liste de tous les 4D Server récemment utilisés. La liste est triée par ordre alphabétique. Pour vous connecter à un serveur depuis cette liste, double-cliquez sur son nom ou sélectionnez-le et cliquez sur le bouton **OK**.

Pour supprimer un serveur de la liste, sélectionnez-le et appuyez sur la touche **Suppr** ou **Ret. Arrière**.

Page "TCP/IP"



4D Server est doté d'un système de publication TCP/IP intégré, qui publie par défaut le nom des bases 4D Server publiées sur le réseau. Ces noms apparaissent dans la page **TCP/IP** de la boîte de dialogue de connexion.

La liste est triée par ordre alphabétique. Pour vous connecter à un serveur depuis cette liste, double-cliquez sur son nom ou sélectionnez-le et cliquez sur le bouton **OK**.

Notes :

- Un accent circonflexe (^) apparaît devant le nom des bases publiées en mode sécurisé. Pour plus d'informations, reportez-vous à la section Crypter les connexions client/serveur.
- Il est possible d'interdire la publication dynamique du nom de la base sur le réseau (cf. section Préférences de publication). Dans ce cas, la connexion doit être effectuée manuellement dans la page "Personnalisée".

Page "Personnalisée"



La page **Personnalisée** permet de désigner, via l'adresse IP du serveur, une base publiée sur le réseau et de lui associer un nom personnalisé.

En effet, le système de publication TCP/IP de 4D Server peut être personnalisé afin que le nom des bases ne soit pas automatiquement publié sur le réseau (cf. section Préférences de publication). Dans ce cas, ces noms n'apparaissent pas dans la liste du client. Il est nécessaire dans ce cas de connaître l'adresse IP du serveur et de la saisir manuellement.

- **Nom de la base** : permet de définir le nom de la base 4D Server. Ce nom sera utilisé dans la page **Récents** pour désigner la base.
- **Adresse réseau** : permet de saisir l'adresse IP de la machine sur laquelle est lancé 4D Server. Si deux serveurs sont exécutés simultanément sur le même poste, l'adresse IP doit être suivie de deux-points et du numéro de port, par exemple : 192.168.92.104:19814. Par défaut, le port de publication d'un 4D Server est le 19813. Ce numéro peut être modifié dans les Préférences de l'application (cf. section Préférences de configuration).

Note : Si une base était sélectionnée dans les pages **Récents** ou **TCP/IP** au moment où vous avez cliqué sur l'onglet **Personnalisée**, les deux champs affichent les informations correspondantes.

Une fois que cette page désigne un serveur, cliquer sur le bouton **OK** vous permet de vous connecter au serveur. Par la suite, le nom du serveur sera listé dans la page **Récents**.

Note : Si la base est publiée en mode sécurisé, il faut impérativement faire débiter son nom par un accent circonflexe (^), sinon la connexion sera refusée. Pour plus d'informations, reportez-vous à la section Crypter les connexions client/serveur.

Forcer la mise à jour des ressources locales

Cette option provoque la mise à jour systématique des ressources locales sur le poste client au moment de la connexion. Les ressources locales sont les informations structurelles relatives à la base stockées sur chaque poste client.

La mise à jour des ressources locales est en principe automatique sur le poste client lorsque la structure de la base a été modifiée depuis la dernière connexion. Dans la plupart des cas, cette option est inutile. Toutefois, dans certains cas spécifiques, il peut être nécessaire de forcer la mise à jour.

Cette option peut également être définie au niveau du poste serveur, elle s'applique dans ce cas à tous les postes clients. Pour plus d'informations, reportez-vous à la section Créer ou ouvrir une base 4D Server.

Vous pouvez créer un fichier de chemin d'accès ("path") qui vous permet d'accéder directement à une base 4D Server depuis le poste client.

Lorsque vous glissez-déposez un fichier de chemin d'accès sur l'application 4D Client (ou double-cliquez sur le fichier), vous accédez directement à la base 4D Server, sans qu'il soit nécessaire de la désigner dans la boîte de dialogue de connexion.

En outre, le fichier de connexion peut contenir le mot de passe utilisateur (crypté). Dans ce cas, l'accès à la base 4D Server est immédiat.

Pour créer un document de Chemin d'accès à une base :

1. Dans la boîte de dialogue d'identification de l'utilisateur de 4D Client, sélectionnez ou saisissez (suivant le paramétrage courant) le nom de l'utilisateur duquel enregistrer l'accès, puis saisissez son mot de passe.

Ces paramètres doivent être valides pour que vous puissiez accéder à la boîte de dialogue d'enregistrement du fichier de connexion.



Note : Cette boîte de dialogue apparaît juste après la sélection de la base 4D Server, lorsque le système de protection par mots de passe est activé.

2. Cliquez sur le bouton Enregistrer.

La boîte de dialogue suivante apparaît :



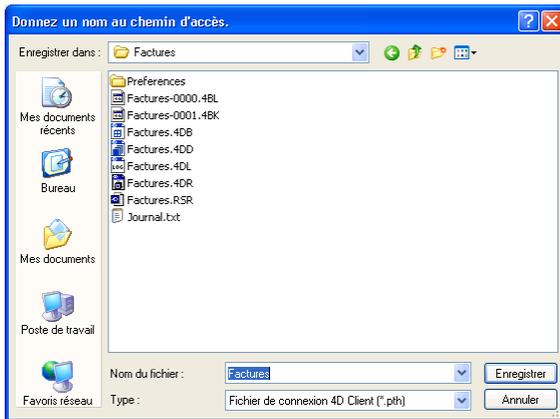
Vous pouvez enregistrer le chemin d'accès avec ou sans mot de passe.

- **Avec mot de passe** : dans ce cas, le chemin d'accès de la base et le mot de passe (crypté) de l'utilisateur sont stockés dans le document. Au lancement de 4D Client, aucune boîte de dialogue n'apparaît, la connexion est immédiate.
- **Sans mot de passe** : dans ce cas, le chemin d'accès de la base est stocké mais l'utilisateur devra saisir son mot de passe pour pouvoir l'ouvrir.

Note : Le bouton **Avec mot de passe** n'apparaît pas si l'option "L'utilisateur peut changer son mot de passe" n'est pas cochée dans la page Application/accès des Préférences de 4D Server.

3. Cliquez sur le bouton Avec mot de passe ou Sans mot de passe.

Une boîte de dialogue standard d'enregistrement de fichier apparaît, vous permettant de désigner le nom et l'emplacement du fichier sur votre disque. L'extension des fichiers de chemin d'accès est ".pth".



4. Choisissez le nom et l'emplacement du fichier et validez la boîte de dialogue d'enregistrement.

Le fichier de connexion 4D Client est stocké sur disque.



MaBase.PTH

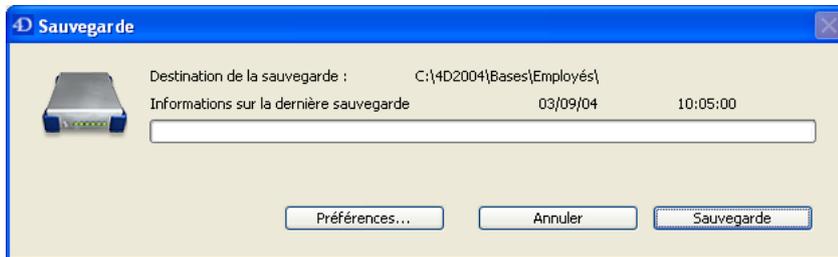
5

Menus de 4D Server



Sauvegarder

Cette commande de menu permet de déclencher une sauvegarde de la base à tout instant. Lorsque vous sélectionnez cette commande, la boîte de dialogue suivante apparaît :



- Le bouton **Sauvegarde** lance immédiatement une sauvegarde en tenant compte des paramètres définis dans la boîte de dialogue des Préférences de l'application (fichiers sauvegardés, emplacement des archives, nombre de jeux, etc.).
- Le bouton **Préférences** ouvre le thème "Sauvegarde" des Préférences, vous permettant de visualiser et éventuellement de modifier les paramètres de sauvegarde courants.
- Le bouton **Annuler** interrompt le processus de sauvegarde.

Pour plus d'informations sur le paramétrage des sauvegardes, reportez-vous au manuel Mode Structure de la documentation de 4e Dimension.

Vérifier le fichier d'historique

Le fichier d'historique stocke toutes les opérations effectuées sur les données de la base (ajout, modification, suppression) — qu'elles aient été effectuées par un utilisateur (à partir de 4D Client), une procédure stockée, un plug-in, un navigateur Web ou encore 4D Open. En cas d'incident sur la base, la restitution de la dernière sauvegarde et l'intégration du fichier d'historique permet de retrouver la base dans l'état exact où elle se trouvait.

La commande **Vérifier le fichier d'historique** affiche une fenêtre permettant de visualiser le contenu du fichier d'historique courant :



The screenshot shows a window titled 'Historique' with a table of database operations. The table has five columns: 'Action', 'Table', 'Utilisateur', 'Date', and 'Heure'. The data is as follows:

	Action	Table	Utilisateur	Date	Heure
17	Ajout	Table 3	Super_Utilisat...	17/02/2004	15:21:04
18	Ajout	Table 3	Super_Utilisat...	17/02/2004	15:21:04
19	Suppression	Table 3	Super_Utilisat...	17/02/2004	15:21:21
20	Suppression	Table 3	Super_Utilisat...	17/02/2004	15:21:21
21	Suppression	Table 3	Super_Utilisat...	17/02/2004	15:21:21
22	Ajout	Table 2	Super_Utilisat...	17/02/2004	15:21:26
23	Ajout	Table 2	Super_Utilisat...	17/02/2004	15:21:28
24	Suppression	Table 2	Super_Utilisat...	17/02/2004	15:21:35
25	Modification	Table 2	Super_Utilisat...	17/02/2004	15:21:39

At the bottom of the window, there are two buttons: 'Annuler' and 'Revenir en arrière'.

Cette fenêtre est utile pour analyser l'utilisation d'une base de données ou détecter la ou les opération(s) à l'origine d'erreurs ou de dysfonctionnements. Elle permet notamment de vérifier les opérations effectuées par chaque poste client.

Cette fenêtre permet également de revenir en arrière parmi les opérations effectuées sur les données de la base (bouton **Revenir en arrière**).

Pour plus d'informations sur ces fonctions, reportez-vous au manuel Mode Utilisation de la documentation de 4e Dimension.

Enregistrer la base courante comme Service

Désenregistrer la base courante

Désenregistrer tous les services serveur

4D Server peut être lancé comme un Service sous Windows et sous Mac OS X.

Une application 4D Server enregistrée comme service est automatiquement lancée au démarrage de la machine avec la base courante, avant même l'ouverture d'une session utilisateur. Elle n'est pas fermée lorsque l'utilisateur quitte sa session.

Ce fonctionnement permet de garantir la disponibilité d'une base 4D Server, même en cas d'incident nécessitant le redémarrage de la machine. La maintenance peut être effectuée à distance.

Note : Pour plus d'informations sur les mécanismes de gestion des Services, reportez-vous à la documentation de votre système d'exploitation.

Pour enregistrer la base de données 4D Server comme un Service, sélectionnez la commande de menu **Enregistrer la base courante comme Service**. Au prochain démarrage de la machine, 4D Server sera automatiquement lancé et la base courante ouverte.

Vous pouvez enregistrer autant de bases de données que vous le souhaitez (mais une seule instance par base).

Important : Veillez à utiliser pour l'ouverture de session un compte d'utilisateur valide et ayant accès à une imprimante, sinon un message d'erreur s'affichera. L'accès aux paramètres des services s'effectue généralement via le panneau de configuration Outils d'administration/Services.

Si vous souhaitez désenregistrer votre base de données courante, sélectionnez **Désenregistrer la base courante** à partir du menu **Fichier** de 4D Server. Cette commande est grisée si la base n'est pas enregistrée comme service.

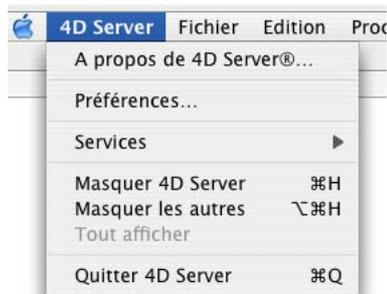
Si vous souhaitez désenregistrer toutes les bases de 4D Server à la fois, sélectionnez **Désenregistrer tous les services serveur** à partir du menu **Fichier** de 4D Server. Cette commande est grisée si aucun service 4D Server n'est actif.

Vous ne pouvez pas désenregistrer 4D Server à partir du menu de 4D Server si l'application a été lancée comme un service au démarrage car les trois commandes de menu sont désactivées. Il faut que vous passiez par le panneau de configuration des Services pour arrêter le services du serveur.

Quitter

Cette commande permet de refermer l'application 4D Server. Pour plus d'informations, reportez-vous à la section Quitter 4D Server.

Note : Sous Mac OS X, la commande **Quitter** se trouve dans le menu **4D Server** (menu application).



Référence

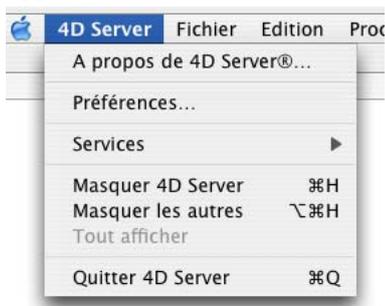
Menu Aide, Menu Données, Menu Edition, Menu Process, Menu Web.

Edition	
Annuler	Ctrl+Z
Couper	Ctrl+X
Copier	Ctrl+C
Coller	Ctrl+V
Effacer	
Tout sélectionner	Ctrl+A
Afficher le Presse-papiers	
Préférences...	

Le menu **Edition** de 4D Server comporte les commandes standard de copier/coller, la commande **Afficher le Presse-papiers**, etc.

Ce menu comporte également (sous Windows) la commande **Préférences...** Cette commande affiche la boîte de dialogue des Préférences de l'application, permettant de paramétrer de nombreux fonctionnements de la base. Pour plus d'informations sur cette boîte de dialogue, reportez-vous au manuel *Mode Structure* de la documentation de 4D.

Note : Sous Mac OS, la commande **Préférences...** est placée dans le menu **4D Server** (menu application).



Référence

Menu Aide, Menu Données, Menu Fichier, Menu Process, Menu Web.

Process	
Tuer	Ctrl+D
Tout déployer	Ctrl+E
Tout contracter	Ctrl+G
Cacher la fenêtre des process Afficher l'Explorateur d'exécution	Ctrl+,
Tracer	Ctrl+T

Tuer

Attention : La commande de menu **Tuer** n'est proposée qu'à des fins de débogage et de gestion des process. Vous ne devez l'utiliser que lorsqu'un client ne peut pas terminer lui-même un process.

L'effet de la commande **Tuer** varie suivant le type de process sélectionné dans la fenêtre de la liste des process :

- Si un **process de connexion** est sélectionné, cette commande tue le process.
- Si la **liste d'un utilisateur** est sélectionnée, cette commande tue tous les process qui appartiennent à cet utilisateur et enlève l'utilisateur de la liste. Cette action déconnecte automatiquement l'utilisateur.
- Si une **procédure stockée** est sélectionnée, la commande stoppe l'exécution de la procédure stockée.
- Si la **liste des procédures stockées** est sélectionnée, la commande stoppe l'exécution de toutes les procédures stockées.
- Si un **process de connexion Web** est sélectionné, la commande tue le process de connexion Web.
- Si la **Liste des clients Web** est sélectionnée, la commande tue tous les process de connexion Web.

Vous pouvez tuer tous les process à l'exception des process du moteur. Si un tel process est sélectionné, la commande **Tuer** est désactivée.

Si vous essayez d'utiliser un process tué sur un 4D Client, un message d'alerte vous indique que le process n'est plus disponible. L'alerte ci-dessous, par exemple, est affichée lorsque la commande de menu **Tuer** a été appliquée à un utilisateur :



Erreur -10001, la connexion à la base de données a été interrompue

Note : Les erreurs de connexion sont décrites dans la section Erreurs réseau du manuel Langage de 4D.

Si un poste client met accidentellement fin à sa connexion au serveur, le serveur s'aperçoit, après quelques secondes, que la connexion avec ce client a été interrompue. Le serveur effectue alors automatiquement les opérations suivantes :

- Libération des enregistrements verrouillés,
- Annulation de toutes les transactions en cours non annulées et non validées,
- Suppression de tous les process de ce client,
- Suppression du client de la liste des clients dans la fenêtre des process de 4D Server.

Par conséquent, il n'est pas nécessaire d'appliquer la commande **Tuer** à l'utilisateur, 4D Server met automatiquement à jour la liste des utilisateurs.

Tout déployer

Tout contracter

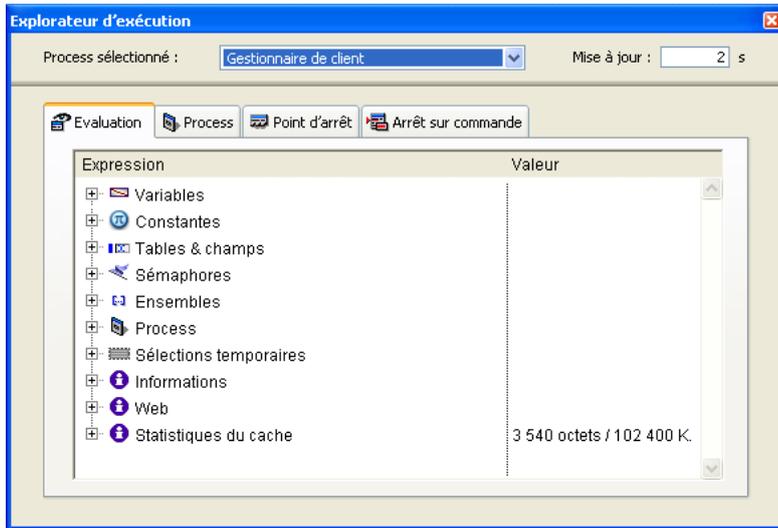
Ces commandes vous permettent de déployer ou contracter les process pour tous les utilisateurs qui apparaissent dans la liste des utilisateurs.

Cacher la fenêtre des process/Afficher la fenêtre des process

Cette commande vous permet d'afficher ou de cacher la fenêtre d'administration (aussi appelée fenêtre des process) de 4D Server.

Afficher l'Explorateur d'exécution/Cacher l'Explorateur d'exécution

Cette commande affiche ou de cache la fenêtre de l'Explorateur d'exécution de 4D Server.



L'Explorateur d'exécution permet de visualiser le comportement des différents éléments structurels de la base et de vérifier que les ressources disponibles sont correctement exploitées. L'Explorateur d'exécution est particulièrement utile en phase de développement et d'analyse d'une base de données.

La fenêtre se compose de quatre pages, accessibles via les onglets correspondants : **Evaluation, Process, Points d'arrêt** et **Arrêt sur commande**. Le fonctionnement de l'Explorateur d'exécution sur 4D Server est identique à celui de 4e Dimension. Pour plus d'informations, reportez-vous au manuel Mode Structure de 4e Dimension.

Tracer

La commande Tracer peut être appliquée aux process suivants :

- **Process de connexion** afin de tracer l'exécution des triggers dans le contexte de ce process
- **Process de connexion Web** afin de tracer le code exécuté sur la machine serveur en réponse à la requête d'un browser Web
- **Procédure stockée** afin de tracer la méthode projet exécutée en tant que procédure stockée.

Sélectionner la commande **Tracer** affiche la fenêtre du débogueur pour le process sélectionné, dès que ce process commence à exécuter du code. Vous pouvez aussi afficher la fenêtre du débogueur pour un process en appelant la commande TRACE à partir d'une méthode exécutée dans ce process.

Le point qui nous intéresse particulièrement ici est que 4D Server “mémorise” l'appel à Trace :

- Si le process est en train d'exécuter du code, le Débogueur apparaît immédiatement pour ce process.
- Si le process n'est pas en train d'exécuter du code (par exemple le process attend un événement particulier pendant la saisie de données), le débogueur apparaîtra lorsque le process reprendra l'exécution de code.

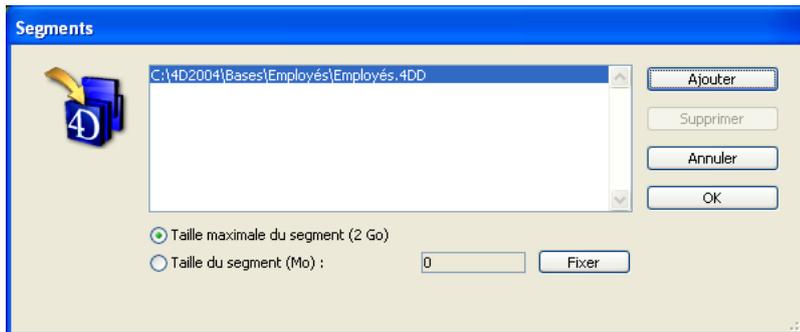
Pour plus d'informations sur la commande TRACE, reportez-vous au manuel Langage de 4e Dimension.

Référence

Menu Aide, Menu Données, Menu Edition, Menu Fichier, Menu Web.



La commande de menu **Nouveau segment...** affiche la fenêtre de gestion des segments suivante :



Vous pouvez segmenter un fichier de données lors de la création de la base ou au cours de son utilisation. Si la taille de votre fichier de données est importante, vous pouvez le segmenter.

La segmentation vous permet de stocker une quantité de données quasiment infinie.

Note : Vous n'avez pas besoin de créer des segments tant que la taille des données de votre base n'atteint pas 2 gigaoctets, sauf limitation de votre disque dur.

Pour segmenter votre fichier de données, vous divisez votre fichier en segments, puis vous indiquez sur quel(s) volume(s) vous voulez les stocker. Par exemple, 4 gigaoctets de données peuvent être divisés en deux segments de 2 Go chacun. La taille de chaque segment peut être fixée afin de préserver de l'espace sur votre disque dur pour d'autres fichiers et d'éviter la saturation.

Si votre fichier de données dépasse les 2 Go, vous devez créer des segments de données pouvant atteindre à leur tour une taille maximale de 2 Go. Il est possible de segmenter des fichiers de données existants sur le poste serveur en appelant la fenêtre Segments.

Pour de plus amples informations sur la segmentation des fichiers de données, reportez-vous au manuel Mode Structure de 4e Dimension.

Référence

AJOUTER SEGMENT DE DONNEES, Menu Aide, Menu Edition, Menu Fichier, Menu Process, Menu Web.



Ce menu permet de démarrer et de stopper le serveur Web de 4D Server à tout moment.

Note : Sous Mac OS, ce menu peut ne pas fonctionner si le serveur Web démarre à l'aide d'une session root temporaire (cf. section Mise en route du serveur Web et gestion des connexions dans le manuel Langage de 4D).

Le serveur Web de 4D peut être démarré de trois manières :

- A partir du menu **Web** dans la barre de menus de 4D Server ou du menu **Exécuter** de 4D Client. Ces menus vous permettent de lancer ou d'arrêter le serveur Web.
- Par publication automatique de la base de données à son ouverture. Pour cela, cochez l'option **Publier la base au démarrage** dans la page "Web>Configuration" de la fenêtre des Préférences de l'application. La base sera alors automatiquement publiée sur le Web lors de chaque ouverture de 4e Dimension ou de 4D Server.
- Par programmation, en appelant la commande LANCER SERVEUR WEB.

Référence

Menu Aide, Menu Données, Menu Edition, Menu Fichier, Menu Process, Mise en route du serveur Web et gestion des connexions, Présentation du serveur Web.



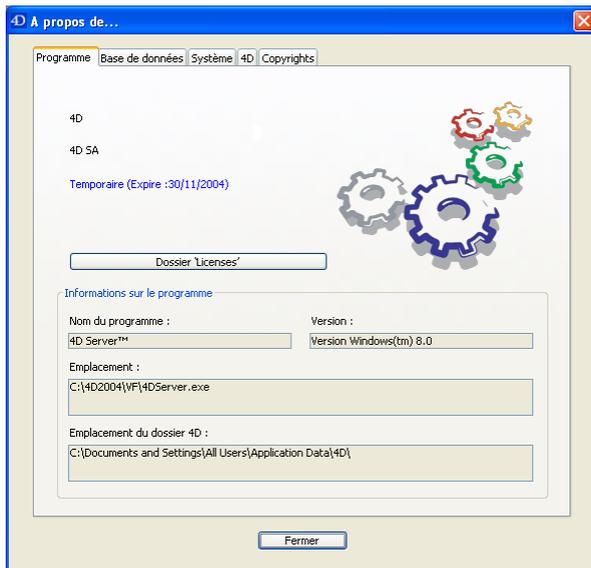
Les premières commandes du menu **Aide** permettent d'afficher et de consulter la documentation de 4D Server sous forme d'aide en ligne.

Mise à jour des licences...

Cette commande de menu affiche la fenêtre permettant d'activer des licences supplémentaires dans votre environnement 4D.

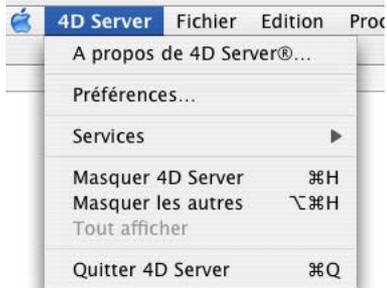
Pour plus d'informations sur cette boîte de dialogue, reportez-vous au Guide d'installation de 4e Dimension.

La commande **A propos de 4D Server...** affiche la fenêtre d'A propos de 4D Server, fournissant diverses informations sur l'application 4D Server (noms et emplacements des fichiers), la base en cours d'exécution et les fonctions système. Ces informations sont classées par pages, accessibles via des onglets :



La page **4D** permet en outre d'accéder à la fonction d'enregistrement en ligne de 4D Server.

Note : Sous Mac OS, la commande **A propos de 4D Server** est placée dans le menu **4D Server** (menu application).



Référence

Menu Données, Menu Edition, Menu Fichier, Menu Process, Menu Web.

6

4D Server et le langage

4D

Avec 4D Server, vous pouvez exécuter du code 4D sur le poste serveur. Il y a trois situations dans lesquelles du code 4D peut être exécuté sur la machine serveur :

- Triggers
- Procédures stockées
- Méthodes base

Triggers

Un trigger est une méthode associée à une table. Les triggers peuvent empêcher des opérations “illégalles” sur les enregistrements de votre base de données. Les triggers sont des outils très puissants qui permettent de restreindre les opérations sur une table ainsi que d'empêcher la perte ou la corruption accidentelle de données. Par exemple, dans un système de facturation, vous pouvez empêcher que ce soit d'ajouter un enregistrement sans indiquer le nom du client.

Les triggers sont exécutés sur la machine sur laquelle le moteur de la base de données est placé. Avec 4D Server, les triggers sont exécutés dans le process du contexte sur la machine serveur, et non pas sur la machine cliente. Avec 4D Server, un trigger est exécuté dans le contexte des process utilisateur qui appellent l'opération de la base. Le trigger n'a cependant pas accès aux variables process du process utilisateur.

Pour plus d'informations sur les triggers, reportez-vous à la section Présentation des triggers dans le manuel Langage de 4D.

Procédures stockées

Une procédure stockée est une méthode projet exécutée dans un process séparé sur le poste serveur (ou, à compter de 4D Server 6.5, sur tout poste client), par opposition au poste client qui est à l'origine de la méthode. Pour plus d'informations sur ce point, reportez-vous à la section Procédures stockées.

Méthodes base

Quatre méthodes base sont exécutées uniquement sur la machine serveur :

- Méthode base Sur démarrage serveur
- Méthode base Sur arrêt serveur
- Méthode base Sur ouverture connexion serveur
- Méthode base Sur fermeture connexion serveur

Quatre autres méthodes base peuvent être exécutées sur la machine serveur ou sur un poste client en fonction du contexte :

- Méthode base Sur authentification Web
- Méthode base Sur connexion Web
- Méthode base Sur démarrage sauvegarde
- Méthode base Sur arrêt sauvegarde

Reportez-vous aux sections correspondantes dans ce manuel ainsi que dans le manuel Langage de 4D pour plus d'informations sur chacun des types de méthodes base.

4D Server et les variables

- 4D Server maintient une table de variables interprocess. La portée de ces variables est le poste serveur. Lorsque 4D Server exécute une base compilée, la définition de la table des variables interprocess est commune entre le serveur et les machines clientes, chaque machine possédant sa propre instance.
- Chaque procédure stockée possède sa propre table de variables process. Lorsqu'une base compilée est exécutée, la table des variables process est commune entre toutes les procédures stockées et les process utilisateur exécutés sur toutes les machines client, chaque process possédant sa propre instance.
- En mode interprété, les méthodes base et triggers peuvent créer et utiliser des variables process de manière dynamique pendant chaque phase de l'exécution. Ceci n'est pas valable en mode compilé. Lorsque vous exécutez une base de données compilée, les méthodes base et les triggers ne partagent qu'une table commune des variables process (dont la définition est identique à celle des autres process).

4D Server et les ensembles

Avec 4D Server, les ensembles interprocess et process sont maintenus sur le poste serveur, tandis que les ensembles locaux sont maintenus sur les postes clients. Pour plus d'informations sur ce point, reportez-vous à la section 4D Server et les ensembles.

Comme décrit dans la section Présentation des ensembles du manuel Langage de 4e Dimension, vous pouvez créer et utiliser des ensembles interprocess, process et locaux :

- **Ensembles process** : Un ensemble process n'est "visible" que par le process dans lequel il a été créé. Les ensembles process sont effacés à la fin de l'exécution de la méthode process. Le nom des ensembles process ne comporte aucun préfixe particulier.
- **Ensembles interprocess** : Un ensemble est interprocess lorsque son nom est préfixé des symboles (<>) — le signe "inférieur à" suivi du signe "supérieur à".

Note : Cette syntaxe est valable sous Windows et Mac OS. De plus, sous Mac OS uniquement, vous pouvez employer le symbole diamant (Option+v sur un clavier français).

- **Ensembles locaux/clients** : Le nom d'un ensemble local/client est préfixé du symbole dollar (\$). A noter que l'ensemble système UserSet, bien que son nom ne débute pas par \$, est un ensemble local/client.

Avec 4D Server, les ensembles interprocess et process sont maintenus sur le poste serveur, alors que les ensembles locaux sont maintenus sur les postes clients.

Conseil : Généralement, il est préférable d'utiliser des ensembles process ou interprocess car leur manipulation est optimisée pour l'architecture client/serveur.

4D Server et les commandes de gestion des ensembles

Ce paragraphe décrit le comportement des commandes de gestion des ensembles dans l'architecture client/serveur, lorsqu'elles sont exécutées sur le poste client :

- **ENSEMBLE VIDE**

L'ensemble vide est créé sur le poste serveur. Un ensemble local est alors copié par le réseau depuis le serveur vers le client. Un ensemble process ou interprocess reste et est maintenu sur le poste serveur.

- **NOMMER ENSEMBLE**
- **CREER ENSEMBLE SUR TABLEAU**

L'ensemble est créé sur le poste serveur. Un ensemble local est alors copié par le réseau depuis le serveur vers le client. Un ensemble process ou interprocess reste et est maintenu sur le poste serveur.

- **UTILISER ENSEMBLE**

Un ensemble local est copié en premier lieu par le réseau du poste client vers le poste serveur, puis est utilisé sur le poste serveur pour modifier la sélection de la table. Un ensemble process ou interprocess est utilisé localement, sur le poste serveur, pour modifier la sélection de la table.

- **STOCKER ENSEMBLE**

Un ensemble local est stocké localement, sur le poste client.

Un ensemble process ou interprocess est copié en premier lieu par le réseau du poste serveur vers le poste client puis est stocké en local sur le poste client.

- **CHARGER ENSEMBLE**

Un ensemble local est chargé du disque du poste client.

Un ensemble process ou interprocess est d'abord chargé du disque du poste client puis est copié par le réseau du poste client vers le poste serveur.

- **EFFACER ENSEMBLE**

- **ADJOINDRE ELEMENT**

- **ENLEVER ELEMENT**

- **Appartient a ensemble**

- **Enregistrements dans ensemble**

Ces cinq commandes accèdent à un ensemble local sur le poste client. Avec un ensemble process ou interprocess, une requête est envoyée par le réseau au poste serveur afin d'obtenir l'information ou d'effectuer l'action.

- **DIFFERENCE**

- **INTERSECTION**

- **REUNION**

Ces commandes nécessitent que les trois ensembles passés en paramètres résident sur le même poste. Par conséquent, tous les ensembles doivent être locaux, ou aucun d'entre eux ne doit l'être.

- **COPIER ENSEMBLE**

A l'aide de **COPIER ENSEMBLE**, vous pouvez copier tout ensemble dans un autre. Par exemple, vous pouvez copier un ensemble local dans un ensemble interprocess ou process. Dans ce cas, l'ensemble est recopié par le réseau du poste client vers le poste serveur.

4D Server et l'ensemble UserSet

4D Client crée l'ensemble UserSet en tant qu'ensemble local (bien que son nom ne débute pas par un \$) afin d'optimiser la création de l'ensemble — issu des actions de l'utilisateur effectuées à la suite d'un MODIFIER SELECTION ou VISUALISER SELECTION.

Si vous souhaitez passer UserSet comme paramètre à une commande DIFFERENCE, INTERSECTION ou REUNION dont les autres paramètres sont des ensembles process ou interprocess, vous devez en premier lieu copier UserSet (un ensemble local) dans un ensemble process ou interprocess puis utiliser l'ensemble résultant avec la commande.

Exemple :

```
TOUT SELECTIONNER ([uneTable])
  ` Permettons à l'utilisateur sélectionner des enregistrements
MODIFIER SELECTION ([uneTable];*)
  ` Vérifions si l'utilisateur a sélectionné des enregistrements
Si (Enregistrements dans ensemble("UserSet")>0)
  ` Recherchons les enregistrements à exclure
  CHERCHER([uneTable];[uneTable]aFlag#0)
  ` Créons un ensemble à partir de la sélection résultante
  NOMMER ENSEMBLE([uneTable];"A exclure")
  Si (Type application = 4D Client)
    ` UserSet est un ensemble local, copions-le dans un ensemble non-local
    COPIER ENSEMBLE ("UserSet";"SélectionUtilisateur") ` ← Copie par le réseau
    ` Appelons DIFFERENCE et passons 3 ensembles non-locaux en paramètres
    DIFFERENCE ("SélectionUtilisateur";"A exclure";"SélectionUtilisateur")
  Sinon
    ` Appelons DIFFERENCE
    DIFFERENCE ("UserSet";"A exclure";"SélectionUtilisateur")
  Fin de si
  EFFACER ENSEMBLE("A exclure")
  UTILISER ENSEMBLE("SélectionUtilisateur")
  EFFACER ENSEMBLE("SélectionUtilisateur")
Fin de si
```

La même opération peut être effectuée avec le code suivant :

```
TOUT SELECTIONNER ([uneTable])
  ` Permettons à l'utilisateur sélectionner des enregistrements
MODIFIER SELECTION ([uneTable];*)
  ` Vérifions si l'utilisateur a sélectionné des enregistrements
Si (Enregistrements dans ensemble("UserSet")>0)
  ` Recherchons les enregistrements à exclure
  CHERCHER([uneTable];[uneTable]aFlag#0)
  Si (Type application = 4D Client)
  ` Créons un ensemble local à partir de la sélection résultante
  NOMMER ENSEMBLE([uneTable];"$A Exclure") ` ← Copie du serveur vers le client
  ` Appelons DIFFERENCE et passons 3 ensembles locaux en paramètres
  DIFFERENCE ("UserSet";"$A Exclure";"UserSet")
  Sinon
  ` Créons un ensemble non-local à partir de la sélection résultante
  NOMMER ENSEMBLE([uneTable];"A Exclure")
  ` Appelons DIFFERENCE
  DIFFERENCE ("UserSet";"A Exclure";"SélectionUtilisateur")
  Fin de si
  EFFACER ENSEMBLE("$A Exclure")
  UTILISER ENSEMBLE("UserSet") ` ← Copie du client vers le serveur
Fin de si
```

Dans le premier exemple, trois ensembles sont créés et un seul est recopié à travers le réseau. Dans le second exemple, deux ensembles sont créés et les deux sont copiés sur le réseau. Vous pourrez donc choisir, en fonction de vos besoins, le type de solution qui vous convient le mieux.

4D Server et l'ensemble LockedSet

L'ensemble LockedSet est un ensemble process créé et maintenu sur le poste serveur.

Référence

4D Server et le langage 4D, COPIER ENSEMBLE, Présentation des ensembles.

La Méthode base Sur démarrage serveur est appelée une fois sur le poste serveur lorsque vous ouvrez une base avec 4D Server. La Méthode base Sur démarrage serveur n'est jamais exécutée dans un environnement autre que 4D Server.

La Méthode base Sur démarrage serveur est l'emplacement idéal pour :

- Initialiser les variables interprocess utilisées pendant toute la session 4D Server.
- Démarrer automatiquement des Procédures stockées à l'ouverture de la base.
- Charger des préférences ou des paramétrages sauvegardé(e)s dans ce but lors de la précédente session 4D Server.
- Empêcher l'ouverture de la base si une condition n'est pas remplie (par exemple, absence de ressources système) par un appel explicite à QUITTER 4D.
- Placer toute action devant être automatiquement effectuée à chaque ouverture de la base.

Si vous souhaitez exécuter du code automatiquement sur un poste client lorsque 4D Client se connecte au serveur, utilisez la Méthode base Sur ouverture.

Référence

Import basé sur les procédures stockées (exemple), Méthode base Sur arrêt serveur, Présentation des méthodes base.

La Méthode base Sur arrêt serveur est appelée une fois sur le poste serveur lorsque vous quittez l'application 4D Server et donc la base. La Méthode base Sur arrêt serveur n'est appelée dans aucun environnement 4D autre que 4D Server.

Pour quitter une application serveur, vous pouvez sélectionner la commande de menu **Quitter** sur le serveur ou appeler la commande QUITTER 4D au sein d'une procédure stockée exécutée sur le serveur.

Lorsque le processus de fermeture de la base a été engagé, 4D effectue les actions suivantes :

- S'il n'y a pas de Méthode base Sur arrêt serveur, 4D Server tue un à un chaque process en cours d'exécution, sans distinction.
- S'il existe une Méthode base Sur arrêt serveur, 4D Server exécute cette méthode dans un nouveau process local. Vous pouvez donc utiliser cette méthode base pour informer les autres process, via la communication interprocess, qu'ils doivent mettre fin à leur exécution. Notez que dans tous les cas 4D Server quittera en fin de compte —la Méthode base Sur arrêt serveur peut effectuer toutes les opérations de nettoyage et de fermeture que vous voulez, mais elle ne peut refuser de quitter, et finira par se terminer.

La Méthode base Sur arrêt serveur est l'emplacement idéal pour :

- Stopper les procédures stockées lancées automatiquement à l'ouverture de la base.
- Sauvegarder (localement, sur disque) des préférences ou des paramètres à réutiliser au début de la session suivante dans la Méthode base Sur démarrage serveur.
- Effectuer toute autre action que vous souhaitez déclencher automatiquement à chaque fois que vous quittez la base.

Important : Si vous utilisez la Méthode base Sur arrêt serveur pour refermer des procédures stockées, tenez compte du fait que le serveur quitte dès la fin de l'exécution de la Méthode base Sur arrêt serveur (et non des procédures stockées). Si des procédures stockées tournent encore à cet instant, elles sont purement et simplement tuées.

Par conséquent, si vous voulez être certain que les procédures stockées se terminent avant que le serveur ne les tue, il faut que la Méthode base Sur arrêt serveur leur signale qu'elles doivent mettre fin à leur exécution (par le test d'une variable interprocess, par exemple) mais aussi qu'elle leur laisse le temps de se refermer (boucle de n secondes ou test d'une autre variable interprocess).

Si vous voulez que du code soit exécuté automatiquement sur un poste client lorsque 4D Client met un terme à sa connexion au serveur, utilisez la Méthode base Sur fermeture.

Référence

Méthode base Sur démarrage serveur, Méthodes, Présentation des méthodes base.

Quand la Méthode base Sur ouverture connexion serveur est-elle appelée ?

La Méthode base Sur ouverture connexion serveur est appelée une fois sur la machine serveur chaque fois qu'un poste client démarre un process de connexion. La Méthode base Sur ouverture connexion serveur n'est appelée que par 4D Server, à l'exclusion de tout autre environnement 4D.

4D Client

Avec 4D Client, la Méthode base Sur ouverture connexion serveur est appelée à chaque fois que :

- 4D Client se connecte (démarrage du process principal)
- 4D Client ouvre l'environnement Structure (démarrage du process de structure)
- 4D Client démarre un process non local (avec la commande Nouveau process)
- Un process non local est démarré par un menu ou par la boîte de dialogue "Exécuter une méthode".

Dans chaque cas, avec 4D Client, deux process démarrent. Un sur la machine client, l'autre sur la machine serveur. Sur la machine client, le process exécute le code et envoie les requêtes à 4D Server. Sur la machine serveur, le process gère l'environnement de base de données du process client (c.-à-d. les sélections courantes pour le process utilisateur) et répond aux requêtes envoyées par le process exécuté sur la machine cliente. C'est la raison pour laquelle le process exécuté sur le serveur est appelé **process de connexion** : il réalise sur la machine serveur, par le réseau, via une connexion, ce que le process s'exécutant sur le poste client effectuerait si vous étiez en configuration monoposte et non en client/serveur.

4D Insider

Lorsque vous vous connectez à 4D Server à partir de 4D Insider, vous démarrez un process de connexion sur la machine serveur pour maintenir un environnement de travail pour 4D Insider. Ce process répond aux requêtes envoyées par 4D Insider.

Application 4D Open

Chaque fois qu'une application utilisant 4D Open ouvre une connexion à 4D Server, un process de connexion démarre sur la machine serveur. Ce process répond aux requêtes envoyées via 4D Open et gère le contexte de base de données de la connexion (c.-à-d. les sélections courantes).

Important : Les connexions Web ne provoquent **pas** l'exécution de la Méthode base Sur ouverture connexion serveur. Lorsqu'un navigateur Web se connecte à 4D Server, la Méthode base Sur authentification Web (si elle existe) et/ou la Méthode base Sur connexion Web sont appelées. Pour plus d'informations, reportez-vous à la description de ces méthodes base dans le manuel Langage de 4e Dimension.

Important : Lors du démarrage d'une procédure stockée, la Méthode base Sur ouverture connexion serveur n'est **pas** appelée. Les procédures stockées sont des process serveur et non des process de connexion. Elles exécutent du code sur la machine serveur mais ne répondent pas aux requêtes échangées par 4D Client (ou d'autres clients) et 4D Server.

Comment la Méthode base Sur ouverture connexion serveur est-elle appelée ?

La Méthode base Sur ouverture connexion serveur est exécutée sur le poste serveur dans le process de connexion qui a provoqué l'appel de la méthode.

Si, par exemple, un 4D Client se connecte à une base 4D Server en mode interprété, il démarre le mode Utilisation, le process de structure ainsi que (par défaut) le process d'inscription du client. La Méthode base Sur ouverture connexion serveur est donc exécutée trois fois de suite. La première fois dans le process de connexion de l'environnement Utilisation (process principal), la deuxième fois dans le process d'inscription du client, et la troisième fois dans le process de connexion de Structure. Si les trois process sont respectivement les 6e, 7e et 8e process démarrés sur la machine serveur, et si vous appelez Numero du process courant dans la Méthode base Sur ouverture connexion serveur, le premier Numero du process courant retourne 6, le deuxième 7 et le troisième 8.

Notez que la Méthode base Sur ouverture connexion serveur s'exécute sur le poste serveur, à l'intérieur du process de connexion sur le serveur. Elle ignore tout du process exécuté sur le client. En outre, au moment où la méthode est appelée, le process de connexion n'est pas encore nommé (INFORMATIONS PROCESS ne retournera pas, à ce moment, le nom du process de connexion).

La Méthode base Sur ouverture connexion serveur n'a pas accès à la table des variables process du process exécuté sur le client. Cette table réside sur le poste client, pas sur le serveur.

Dans le cas d'une base de données en mode interprété, lorsque la Méthode base Sur ouverture connexion serveur accède à une variable process, elle travaille avec une table de variables process particulière, créée dynamiquement pour le process de connexion. Comme la Méthode base Sur fermeture connexion serveur sera finalement appelée dans le même process de connexion, vous pourriez imaginer qu'il est possible de maintenir les informations entre les deux méthodes avec des variables process. Cela ne fonctionnera pas en mode compilé.

Dans une base compilée, la Méthode base Sur ouverture connexion serveur partage, avec d'autres méthodes base et avec les triggers, une table de variables process commune maintenue sur la machine serveur. Cette architecture a deux objectifs : permettre l'exécution du code compilé, et réduire la consommation de mémoire. D'abord, vous pouvez accéder à toutes les variables process dans une méthode base ou dans un process. Il suffit que les variables process « y soient ». Ensuite, la création d'une table process pour chaque méthode base ou trigger serait consommatrice de mémoire et de temps d'initialisation. Conclusion : NE VOUS FIEZ PAS aux variables process lors de l'exécution de la Méthode base Sur ouverture connexion serveur et Méthode base Sur fermeture connexion serveur. Utilisez des données stockées dans les variables interprocess ou dans une table.

4D Server passe trois paramètres de type Entier long à la Méthode base Sur ouverture connexion serveur et attend un résultat Entier long. La méthode doit donc être explicitement déclarée avec trois paramètres Entier long ainsi qu'un retour de fonction Entier long :

C_ENTIER LONG(\$0;\$1;\$2;\$3)

Si vous ne retournez pas de valeur dans \$0 et donc laissez la variable indéfinie ou initialisée à zéro, 4D Server estime que la méthode base accepte la connexion. Si vous n'acceptez pas la connexion, retournez une valeur non nulle dans \$0.

Le tableau ci-dessous détaille les informations fournies par les trois paramètres passés à la méthode base :

Paramètre	Description
\$1	Numéro d'utilisateur utilisé en interne par 4D Server pour identifier les utilisateurs
\$2	Numéro de connexion utilisé en interne par 4D Server pour identifier une connexion
\$3	Numéro d'identification du protocole réseau utilisé en interne par 4D Server

Note : A compter de la version 6.8 de 4D Server, le paramètre \$3 retourne toujours 2 ou 29.

Ces trois numéros de référence ne sont pas directement utilisables en tant « sources d'information » à passer, par exemple, comme paramètres à une commande 4D. Ils vous fournissent un moyen d'identifier de manière unique un process de connexion entre la Méthode base Sur ouverture connexion serveur et la Méthode base Sur fermeture connexion serveur. La combinaison de ces trois valeurs est unique à tout moment d'une session 4D Server. Si vous stockez cette information dans une table ou un tableau interprocess, les deux méthodes base peuvent échanger des informations. Dans l'exemple présenté à la fin de cette section, les deux méthodes base utilisent cette information pour stocker l'heure et la date du début et de la fin d'une connexion dans le même enregistrement d'une table.

Exemples

(1) L'exemple suivant montre comment maintenir un historique des connexions à la base de données en utilisant la Méthode base Sur ouverture connexion serveur et la Méthode base Sur fermeture connexion serveur. La table [Server Log] (ci-dessous) sert à garder la trace des process de connexion :

Server Log	
Log ID	L
Log Date	D
Log Time	H
Exit Date	D
Exit Time	H
User ID	L
Connection ID	L
NC ID	L
Process ID	L
Process Name	A

L'information stockée dans cette table est gérée par la Méthode base Sur ouverture connexion serveur et la Méthode base Sur fermeture connexion serveur listées ci-dessous :

```
  ` Méthode base Sur ouverture connexion serveur
C_ENTIER LONG($0;$1;$2;$3)
  ` Créer un enregistrement [Server Log]
CREER ENREGISTREMENT([Server Log])
[Server Log]Log ID:=Numerotation automatique([Server Log])
  ` Enregistrer l'historique Date et Heure
[Server Log]Log Date:=Date du jour
[Server Log]Log Time:=Heure courante
  ` Enregistrer l'information sur la connexion
[Server Log]User ID:=$1
[Server Log]Connection ID:=$2
[Server Log]NC ID:=$3
STOCKER ENREGISTREMENT([Server Log])
  ` Ne retourne pas d'erreur, pour continuer la connexion
$0:=0
```

```

  ` Méthode base Sur fermeture connexion serveur
C_ENTIER LONG($1;$2;$3)
  ` Chercher l'enregistrement [Server Log]
CHERCHER([Server Log];[Server Log]User ID=$1;*)
CHERCHER([Server Log]; & ;[Server Log]Connection ID=$2;*)
CHERCHER([Server Log]; & ;[Server Log]NC ID=$3;*)
CHERCHER([Server Log]; & ;[Server Log]Process ID=0)
  ` Enregistrer date et heure de déconnexion
[Server Log]Exit Date:=Date du jour
[Server Log]Exit Time:=Heure courante
  ` Enregistrer informations process
[Server Log]Process ID:=Numero du process courant
INFORMATIONS PROCESS([Server Log]Process ID;$vsProcName;$vlProcState;
                                                                    $vlProcTime)

[Server Log]Process Name:=$vsProcName
STOCKER ENREGISTREMENT([Server Log])

```

Voici quelques entrées dans [Server Log] montrant plusieurs connexions 4D Client :

Log ID	Log Date	Log Time	Exit Date	Exit Time	User ID	Connection ID	NC ID	Process ID	Process Name
1	21/10/99	16:37:4	21/10/99	18:37:18	35719024	35684424	2	7	Process principal
2	21/10/99	16:37:5	21/10/99	18:37:18	35719024	35687488	2	6	Arnaud
3	21/10/99	16:37:6	21/10/99	18:37:19	35719024	35690552	2	5	Process de structur
4	21/10/99	16:38:0	21/10/99	16:39:16	35719024	35684424	2	7	Process principal
5	21/10/99	16:38:2	21/10/99	16:39:16	35719024	35687488	2	6	Phil Defer
6	21/10/99	16:38:4	21/10/99	16:39:16	35719024	35690552	2	5	Process de structur
7	21/10/99	16:39:4	21/10/99	17:40:09	32376768	32335992	1	8	Process principal
8	21/10/99	16:39:9	21/10/99	17:40:09	32376768	32339056	1	9	Frederique
9	21/10/99	16:39:10	21/10/99	17:40:10	32376768	32342120	1	10	Process de structur

(2) L'exemple suivant interdit toute nouvelle connexion entre 2 et 4 heures du matin.

```

  ` Méthode base Sur ouverture connexion serveur
C_ENTIER LONG($0;$1;$2;$3)

Si((?02:00:00<=Heure courante)&(Heure courante<?04:00:00?))
  $0:=22000
Sinon
  $0:=0
Fin de si

```

Référence

Méthode base Sur fermeture connexion serveur, Présentation des méthodes base.

La Méthode base Sur fermeture connexion serveur est exécutée sur le poste serveur à chaque fois qu'un process de connexion est refermé.

Comme pour la Méthode base Sur ouverture connexion serveur, 4D Server passe trois paramètres de type Entier long à la Méthode base Sur fermeture connexion serveur. En revanche, 4D Server n'attend pas de résultat en retour.

Par conséquent, la méthode doit contenir la déclaration explicite des trois paramètres Entier long :

C_ENTIER LONG(\$1;\$2;\$3)

Le tableau suivant détaille les informations fournies par les trois paramètres passés à la méthode base :

Paramètre	Description
\$1	Numéro d'utilisateur utilisé en interne par 4D Server pour identifier les utilisateurs
\$2	Numéro de connexion utilisé en interne par 4D Server pour identifier une connexion
\$3	Numéro de protocole réseau utilisé en interne par 4D Server

Note : A compter de la version 6.8 de 4D Server, le paramètre \$3 retourne toujours 2 ou 29.

La Méthode base Sur fermeture connexion serveur est le pendant inverse de la Méthode base Sur ouverture connexion serveur. Pour plus d'informations sur ce point, ainsi que pour la description des **process de connexion**, reportez-vous à la description de cette méthode base.

Exemple

Reportez-vous au premier exemple de la Méthode base Sur ouverture connexion serveur.

Référence

Méthode base Sur ouverture connexion serveur, Présentation des méthodes base.

7

Procédures stockées

Qu'est-ce qu'une procédure stockée dans un système SQL ?

L'expression "Procédure Stockée" provient du monde des serveurs SQL. Lorsqu'une station cliente envoie une requête à un serveur SQL, elle envoie en réalité du texte en langage SQL au serveur SQL. Cette requête fait l'objet d'une analyse syntaxique (*parsing*) et est interprétée sur le serveur SQL avant d'être exécutée. De toute évidence, si le texte de la requête est important et si la requête est envoyée plusieurs fois au cours d'une session, il peut y avoir beaucoup de temps passé pour l'envoi de ce code sur le réseau, le parsing et l'interprétation de la requête. Ainsi est née l'idée de faire en sorte que la requête soit envoyée sur le réseau, analysée et interprétée en une seule fois, et qu'ensuite elle soit exécutée à chaque fois qu'elle est envoyée par la station cliente. La solution était de conserver le code source de la requête (en d'autres termes, une procédure) sur le serveur et que le client envoie une requête comportant seulement le nom de la procédure à exécuter. En conséquence, cette procédure est dite "stockée" sur le serveur, d'où le nom de Procédure Stockée. Notez qu'une procédure SQL est une procédure qui peut recevoir des paramètres venant de la station cliente, exécuter les tâches pour lesquelles elle est faite (de façon synchrone ou asynchrone) et le cas échéant retourner un résultat au client. Quand un client demande l'exécution d'une procédure stockée, ce qu'il fait, c'est dans une certaine mesure déléguer l'exécution du code sur le serveur.

Qu'est-ce qu'une procédure stockée dans 4D Server ?

Bien que nous utilisions aussi le nom de procédures stockées puisqu'il est compréhensible par tous dans l'industrie, les objets que nous appelons ainsi dans 4D Server ont des fonctionnalités plus larges que celles que l'on trouve habituellement.

Avec 4D, lorsque par exemple vous utilisez la commande Nouveau process, vous pouvez ouvrir un process utilisateur dans lequel vous allez exécuter une méthode. Cette méthode est appelée une **méthode process** (voir à ce sujet la section Méthodes projet dans le manuel Langage de 4e Dimension).

Avec 4D Server, sur une machine 4D Client, vous pouvez faire la même chose. De plus, en utilisant la commande Exécuter sur serveur, vous pouvez démarrer sur le serveur un process utilisateur dans lequel vous allez exécuter une méthode. En outre, en utilisant la commande EXECUTER SUR CLIENT, vous pouvez exécuter une méthode dans un autre process sur un autre poste client.

Dans les deux cas, cette méthode est appelée une **procédure stockée**, et par extension, le process démarré sur le serveur ou sur un autre poste client est aussi appelé procédure stockée.

La différence principale entre une procédure stockée SQL et une procédure stockée 4D Server est que dans le premier cas vous exécutez une procédure SQL et dans le second, vous exécutez un process 4D.

Architecture des procédures stockées de 4D

Comme un process 4D normal, une procédure stockée dispose de son propre environnement :

- **Une sélection courante par table** : chaque procédure stockée a sa propre sélection courante. Une table peut avoir une sélection courante différente dans chaque procédure stockée.
- **Un enregistrement courant par table** : chaque table peut avoir un enregistrement courant différent dans chaque procédure stockée.
- **Variables** : chaque procédure stockée a ses propres variables process. Les variables process ne sont reconnues que dans le contexte de la procédure stockée à laquelle elles appartiennent.
- **Table par défaut** : chaque procédure stockée a sa propre table par défaut.
- **Ensembles process** : chaque procédure stockée a ses propres ensemble process.
- **Appeler sur erreur** : chaque procédure stockée dispose d'une méthode de gestion d'erreur.
- **Fenêtre de débogage** : chaque procédure stockée dispose de sa propre fenêtre de débogage.

En termes d'interface utilisateur, une procédure stockée peut ouvrir des fenêtres et afficher des données (i.e. AFFICHER ENREGISTREMENT).

Une procédure stockée exécutée sur un ou plusieurs postes 4D Client permet la saisie de données.

En revanche, une procédure stockée exécutée sur le serveur ne permet pas la saisie de données.

Vous pouvez démarrer autant de procédures stockées que vous le souhaitez dans la mesure où votre matériel et la mémoire le permettent. En fait, la machine serveur doit être considérée comme une machine qui est non seulement capable de répondre à des clients 4D et à des browsers Web, mais aussi capable d'exécuter des process qui peuvent interagir avec d'autres process fonctionnant aussi bien sur le serveur que sur des 4D Client.

De la même façon que 4D et 4D Client fournissent un environnement multitâche aux process utilisateurs fonctionnant sur la station cliente, 4D Server fournit un environnement multitâche aux procédures stockées. Par exemple, 4D Server maintient une table des variables interprocess qui peuvent être utilisées par les procédures stockées pour communiquer entre elles.

Que peut faire une procédure stockée ?

Une grande partie fonctionnalités des process et des commandes décrites dans le manuel *Langage* de 4e Dimension s'appliquent aussi aux procédures stockées — à l'exception de la saisie de données pour les procédures stockées exécutées sur le serveur.

Une procédure stockée peut ajouter, rechercher, trier, mettre à jour ou détruire des enregistrements. Une procédure stockée peut utiliser des ensembles et des sélections temporaires, accéder à des documents sur disque, travailler avec des BLOBs, etc. Pensez simplement qu'au lieu de fonctionner sur votre poste 4D Client, vous fonctionnez sur le serveur ou sur un ou plusieurs autres postes clients.

Un avantage évident des procédures stockées exécutées sur le serveur est que précisément une procédure stockée s'exécute sur la machine serveur, là où se trouve le moteur de la base de données. Par exemple, un APPLIQUER A SELECTION n'est pas efficace sur le réseau, mais l'est à l'intérieur d'une procédure stockée. L'exemple proposé dans la section Import basé sur les procédures stockées (exemple) montre les optimisations remarquables que vous pouvez obtenir en utilisant une procédure stockée.

Les procédures stockées exécutées sur un ou plusieurs autres postes clients autorisent, quant à elles, l'optimisation de la répartition des tâches entre les clients, ou encore la communication inter-clients. La description de la commande INSCRIRE CLIENT fournit un exemple simple de messagerie utilisant des procédures stockées exécutées sur les clients.

Cependant, l'avantage le plus évident de l'architecture des procédures stockées se trouve dans la nouvelle dimension qu'elles apportent à 4D Server : grâce aux procédures stockées, vous pouvez implémenter vos propres services 4D Server. La limite, c'est votre imagination. L'exemple fourni dans la section Services basés sur les procédures stockées (exemple) montre une procédure stockée capable de fournir des informations sur 4D Server à ses clients. Vous pouvez, par exemple, lister les volumes de la machine serveur. Cet exemple pourrait être étendu facilement, pour, notamment, renvoyer les informations sur les répertoires ou documents au client.

Que ne peut pas faire une procédure stockée (exécutée sur le serveur) ?

De manière générale, les procédures stockées exécutées sur le serveur ne doivent pas effectuer d'opérations impliquant des éléments d'interface (menus, fenêtres, formulaires...). En effet, les mécanismes de gestion d'interface ne sont pas gérés sur le serveur. Les commandes provoquant l'apparition de boîtes de dialogue sur le poste serveur ainsi que celles liées à la saisie de données sont également à proscrire.

Voici la liste des commandes ne devant PAS être utilisées dans le cadre des procédures stockées exécutées sur le serveur. Ces commandes sont classées en trois catégories :

- **Les commandes interdites sur le serveur**

La présence d'une de ces commandes dans une procédure stockée provoque l'apparition d'une boîte de dialogue d'alerte indiquant que la commande ne peut pas être exécutée sur 4D Server. L'erreur 67 est retournée, elle peut être interceptée à l'aide d'une méthode installée par la commande APPELER SUR ERREUR.

ACTIVER LIGNE MENU
AFFICHER BARRE DE MENUS
AJOUTER ENREGISTREMENT
AJOUTER LIGNE MENU
AJOUTER SOUS ENREGISTREMENT
APPELER PROCESS
APPELER SUR EVENEMENT
CACHER BARRE DE MENUS
CHANGER LICENCES
CHANGER PRIVILEGES
CHANGER RACCOURCI CLAVIER
CHANGER STYLE LIGNE MENU
CHANGER TEXTE LIGNE MENU
CHERCHER PAR EXEMPLE
Creer fenetre externe
CREER FICHIER DONNEES
CUMULER SUR
ECRIRE IMAGE DANS BIBLIOTHEQUE
FILTRER EVENEMENT
GRAPHE SUR SELECTION
IMPRESSION ECRAN
IMPRIMER ENREGISTREMENT
IMPRIMER ETIQUETTES
IMPRIMER LIGNE
IMPRIMER SELECTION
INACTIVER LIGNE MENU
INSERER LIGNE MENU
Marque ligne menu
MARQUER LIGNE MENU
Menu choisi
MODIFIER ENREGISTREMENT
MODIFIER SELECTION
MODIFIER SOUS ENREGISTREMENT
Niveau
NIVEAUX DE RUPTURES
Nombre de lignes de menu
Nombre de menus
OUVRIR FICHIER DONNEES

Page impression
PARAMETRES IMPRESSION
QR ETAT
Raccourci clavier
SAUT DE PAGE
Sous total
Style ligne menu
SUPPRIMER IMAGE DANS BIBLIOTHEQUE
SUPPRIMER LIGNE MENU
Texte ligne menu
Titre menu
UTILISER PARAMETRES IMPRESSION
VISUALISER SELECTION

• **Les commandes déconseillées sur le serveur**

L'utilisation de ces commandes dans des procédures stockées est fortement déconseillée car leur fonctionnement n'est pas adapté à une exécution sur le serveur. Ces commandes peuvent bloquer le serveur, provoquer des erreurs, et plus généralement ne produisent pas les effets escomptés. Aucun code d'erreur spécifique n'est retourné.

Activation
ACTIVER BOUTON
AFFICHER FENETRE
AJOUTER A LISTE
AJOUTER A PRESSE PAPIERS
AJOUTER SEGMENT DE DONNEES
ALLER A CHAMP
ALLER A PAGE
Ancien
Appartient au groupe
Appel exterieur
Apres
Avant
BLOB VERS DOCUMENT
BLOB vers liste
CACHER BARRE OUTILS
CACHER FENETRE
CACHER PROCESS
CHANGER COORDONNEES FENETRE
CHANGER ELEMENT
CHANGER JEU DE CARACTERES
CHANGER MOT DE PASSE
CHANGER POINTEUR SOURIS
CHANGER PROPRIETES ELEMENT
CHANGER PROPRIETES LISTE

CHANGER STYLE
CHANGER TAILLE
CHANGER UTILISATEUR COURANT
Charger liste
CHERCHER PAR FORMULE (2)
Chercher fenetre
CHERCHER (2)
CHOIX COULEUR
CHOIX ENUMERATION
CHOIX FILTRE SAISIE
CHOIX FORMATAGE
CHOIX SAISSABLE
CHOIX VISIBLE
COORDONNEES ECRAN
COORDONNEES FENETRE
Copier liste
Creer document (1)
Creer fichier ressources (1)
C_GRAPHE
DEPLACER FENETRE
DEPLACER OBJET
DERNIERE PAGE
Desactivation
DIALOGUE
Ecran principal
ECRIRE IMAGE DANS PRESSE PAPIERS
ECRIRE TEXTE DANS PRESSE PAPIERS
Ecrire proprietes groupe
Ecrire proprietes utilisateur
EFFACER FENETRE
EFFACER PRESSE PAPIERS
Element parent
Elements selectionnes
En entete
En pied
En rupture
Evenement formulaire
EXPORTER DONNEES (1)
Fenetre premier plan
Fenetre suivante
FILTRER FRAPPE CLAVIER
FIXER COULEURS RVB
FIXER FICHER HISTORIQUE
FIXER MINUTEUR

FIXER PROFONDEUR ECRAN
FIXER TITRES CHAMPS
FIXER TITRES TABLES
FORMULAIRE ENTREE
FORMULAIRE SORTIE
Frappe clavier
GENERER CLIC
GENERER EVENEMENT
GENERER FRAPPE CLAVIER
Hauteur barre de menus
Hauteur ecran
IMPORTER DONNEES (1)
INACTIVER BOUTON
INFORMATION ELEMENT
INSCRIRE CLIENT
INSERER ELEMENT
INVERSER FOND
Largeur ecran
LIRE IMAGE DANS PRESSE PAPIERS
LIRE LISTE GROUPE
LIRE LISTE UTILISATEURS
LIRE PRESSE PAPIERS
LIRE PROPRIETES ELEMENT
LIRE PROPRIETES FORMULAIRE
LIRE PROPRIETES GROUPE
LIRE PROPRIETES LISTE
LIRE PROPRIETES UTILISATEUR
LIRE RECT OBJET
Lire texte dans presse papiers
Lire texte edite
LISTE DES POLICES
LISTE FENETRES
LISTE VERS BLOB
Liste existante
Majuscule enfoncee
MARQUER ENREGISTREMENTS
MAXIMISER FENETRE
MINIMISER FENETRE
Modifie
MONTRER PROCESS
NE PAS VALIDER
Nom de police
Nombre ecrans
Nombre elements

Nouvelle liste
Numero de police
Objet focus
Ouvrir document (1)
Ouvrir fichier ressources (1)
PAGE PRECEDENTE
PAGE SUIVANTE
Page formulaire courante
PARAMETRES DU GRAPHE
PASSER AU PREMIER PLAN
Pendant
Pop up menu
POSITION SOURIS
Position déposer
Position element liste
PREMIERE PAGE
Process de la fenetre
Process de premier plan
PROFONDEUR ECRAN
PROPRIETES GLISSER DEPOSER
REDESSINER
REDESSINER FENETRE
REDESSINER LISTE
REFUSER
SELECTIONNER ELEMENTS PAR POSITION
SELECTIONNER ELEMENTS PAR REFERENCE
SELECTIONNER TEXTE
Selectionner dossier
Self
STOCKER LISTE
SUPPRIMER ELEMENT
SUPPRIMER LISTE
SUPPRIMER UTILISATEUR
Table du formulaire courant
Tester presse papiers
TEXTE SELECTIONNE
TITRE BOUTON
Titre fenetre
TRIER LISTE
TRIER (2)
Type fenetre
Utilisateur courant
Utilisateur supprime

VALIDER

Valider mot de passe

- (1) Uniquement lorsque le premier paramètre est une chaîne vide.
- (2) Uniquement lorsque la syntaxe utilisée provoque l'apparition de la boîte de dialogue (ex : TRIER ([Table])).

• Les commandes sans effet sur le serveur

Ces commandes sont sans effet lorsqu'elles sont exécutées dans une procédure stockée sur le serveur. Aucun code d'erreur spécifique n'est retourné.

CHANGER BARRE

LAISSER MESSAGES

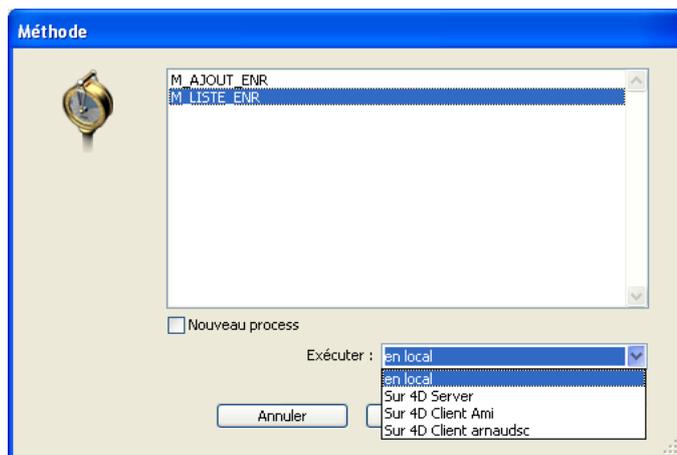
SUPPRIMER MESSAGES

GRAPHE

AFFICHER BARRE OUTILS

Comment lancer une procédure stockée ?

- Avec 4D Client, vous pouvez lancer manuellement une procédure stockée dans la boîte de dialogue d'exécution de méthode :



Vous pouvez l'exécuter sur 4D Server ou sur un ou plusieurs des autres 4D Client. Notez que, pour que les 4D Client apparaissent dans cette liste, ils doivent auparavant avoir été inscrits (cf. section Inscrire les clients et commande INSCRIRE CLIENT).

- Toujours sur 4D Client, vous pouvez lancer une procédure stockée par programmation à l'aide des commandes Executer sur serveur ou EXECUTER SUR CLIENT.

Note : Il n'est pas possible d'utiliser les commandes de gestion des process ENDORMIR PROCESS, SUSPENDRE PROCESS et REACTIVER PROCESS à partir de 4D Client avec des procédures stockées sur le serveur.

- Une méthode exécutée sur 4D Server (Méthode base du serveur ou procédure stockée) peut elle-même lancer une procédure stockée à l'aide des commandes Executer sur serveur, Nouveau process ou EXECUTER SUR CLIENT.

A propos de la communication interprocess entre les procédures stockées et les process utilisateurs

Les procédures stockées peuvent communiquer entre elles à l'aide des :

- variables interprocess
- sémaphores globaux ou locaux
- enregistrements
- ensembles interprocess et sélections temporaires interprocess
- commandes LIRE VARIABLE PROCESS, ECRIRE VARIABLE PROCESS et VARIABLE VERS VARIABLE.

Reportez-vous aux sections correspondantes du manuel *Langage* de 4e Dimension. Une fois encore, gardez à l'esprit que les commandes 4D agissent sur le poste qui exécute la procédure stockée (serveur ou clients) de la même manière qu'en local sur un poste client.

Note : Les mécanismes APPELER PROCESS et Appel extérieur n'ont pas de signification sur le poste serveur car les procédures stockées ne disposent pas d'interface utilisateur avec saisie de données.

En outre, notez cette fonctionnalité importante : les process utilisateur des clients (process tournant sur un poste client) peuvent lire et écrire les variables process (*) d'une procédure stockée à l'aide des commandes LIRE VARIABLE PROCESS, ECRIRE VARIABLE PROCESS et VARIABLE VERS VARIABLE.

(*) ainsi que les variables interprocess du poste serveur.

Important : La communication process "Intermachine" permise par les commandes LIRE VARIABLE PROCESS, ECRIRE VARIABLE PROCESS et VARIABLE VERS VARIABLE n'est possible que du client vers le serveur. C'est toujours un process client qui lit ou écrit les variables d'une procédure stockée.

Référence

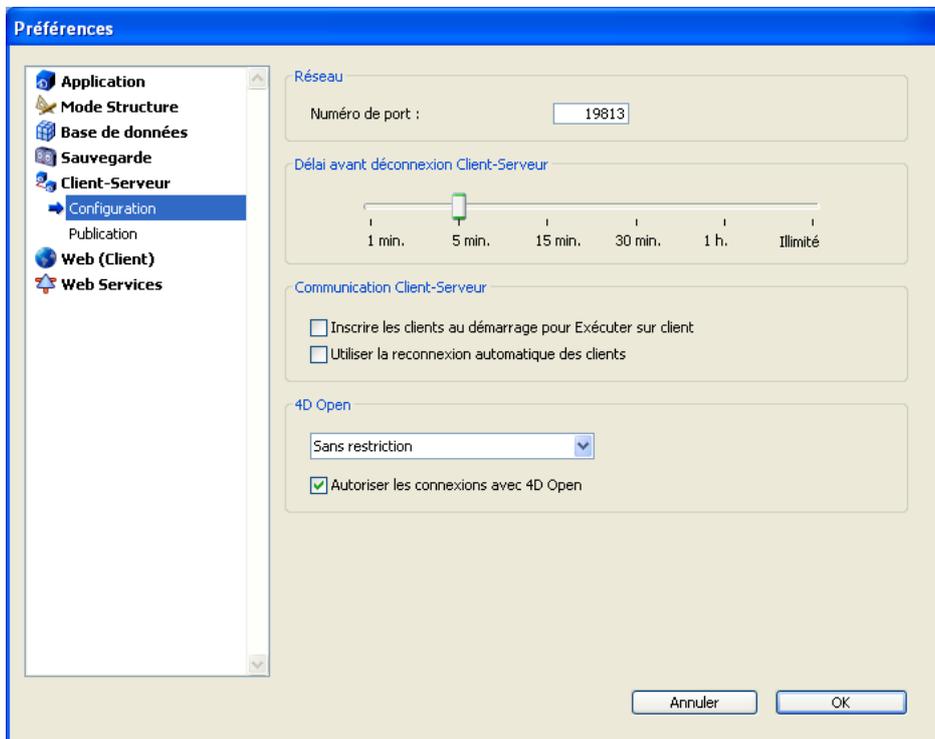
Import basé sur les procédures stockées (exemple), Services basés sur les procédures stockées (exemple).

Vous pouvez exécuter des procédures stockées sur un ou plusieurs autres 4D Client. Le fonctionnement général des procédures stockées exécutées sur les clients est identique à celui des procédures stockées exécutées sur le serveur, à la différence près qu'une procédure stockée exécutée sur un 4D Client peut, elle, "saisir" des données. Ce fonctionnement est détaillé dans la section Procédures stockées.

En outre, tout poste client devant exécuter des procédures à la demande du serveur ou d'un autre poste client doit être explicitement "inscrit" pour la session. Pour inscrire un 4D Client, vous disposez de deux solutions : inscrire automatiquement chaque client à la connexion, ou inscrire les clients par programmation.

Inscrire automatiquement chaque 4D Client qui se connecte à 4D Server

L'option "Inscrire chaque client à la connexion" est disponible dans les Préférences de l'application, page "Configuration" du thème "Client-Serveur" :



Lorsque cette option est cochée, chaque 4D Client qui se connecte à la base est automatiquement référencé auprès de 4D Server comme pouvant exécuter des procédures stockées. Un process portant le nom de la machine du poste client est créé sur le serveur, dans le groupe de process de l'utilisateur. Un process est également créé sur chaque poste client.

Inscrire les 4D Client par programmation

Vous pouvez également inscrire un ou plusieurs 4D Client par programmation. Cette option vous permet de sélectionner les postes clients devant être inscrits et de choisir leur nom d'inscription.

Le thème "Process" comporte la commande INSCRIRE CLIENT, permettant d'inscrire un poste client sous le nom que vous voulez.

Désinscrire un 4D Client

Quel que soit le mode d'inscription des postes clients, vous pouvez à tout moment désinscrire tout poste client du serveur pour la session courante. Pour cela, il vous suffit d'exécuter la commande DESINSCRIRE CLIENT (thème "Process") sur le(s) poste(s) client(s) à désinscrire.

Le process de gestion de l'inscription (portant le nom du client) disparaît alors du groupe de process de l'utilisateur sur le poste serveur, ainsi que sur le poste client.

Note : La commande LIRE CLIENTS INSCRITS vous permet d'obtenir la liste et la charge de travail (le nombre de méthodes restant à exécuter) des clients inscrits dans la session.

Pour plus d'informations sur ces commandes, reportez-vous au manuel *Langage* de 4D.

Référence

DESINSCRIRE CLIENT, INSCRIRE CLIENT, LIRE CLIENTS INSCRITS.

L'exemple suivant démontre comment l'import de données peut être accéléré de manière spectaculaire en environnement client/serveur. La méthode Import classique listée ci-dessous vous permet de mesurer combien de temps prend un import d'enregistrements basé sur la commande LECTURE ASCII :

```
` Méthode projet Import classique
$vhDocRef:=Ouvrir document("")
Si (OK=1)
  FERMER DOCUMENT($vhDocRef)
  FORMULAIRE ENTREE([Table1];"Import")
  $vhStartTime:=Heure courante
  LECTURE ASCII([Table1];Document)
  $vhEndTime:=Heure courante
  ALERTE("L'opération a duré "+Chaine(0+($vhEndTime-$vhStartTime))+ " secondes.")
Fin de si
```

Avec l'import de données classique, 4D Client analyse le fichier ASCII puis, pour chaque enregistrement, crée un nouvel enregistrement, remplit les champs avec les valeurs importées et envoie l'enregistrement au poste serveur afin qu'il soit ajouté à la base. Par conséquent, de nombreuses requêtes circulent sur le réseau. Afin d'optimiser l'opération, vous pouvez utiliser des procédures stockées pour effectuer l'import localement sur le poste serveur. Le poste client charge le document dans un BLOB et déclenche une procédure stockée en passant le BLOB comme paramètre. La procédure stockée place le BLOB dans un document sur le disque du poste serveur, puis importe le document en local. L'import des données est par conséquent effectué localement à une vitesse comparable à celle d'une version monoposte de 4D, car la plupart des requêtes transitant sur le réseau ont été éliminées. Voici la méthode projet CLIENT IMPORT. Lancée sur le poste client, elle déclenche l'exécution de la procédure stockée SERVER IMPORT, listée à la suite :

```
` Méthode projet CLIENT IMPORT
` CLIENT IMPORT ( Pointeur ; Alpha )
` CLIENT IMPORT ( -> [Table] ; Formulaire entrée )

C_POINTEUR($1)
C_ALPHA(31;$2)
C_HEURE($vhDocRef)
C_BLOB($vxData)
C_ENTIER LONG(spErrCode)
```

```

    ` Sélectionnez le document à importer
$vhDocRef:=Ouvrir document("")
Si (OK=1)
    ` Si un document était sélectionné, ne pas le garder ouvert
FERMER DOCUMENT($vhDocRef)
$vhStartTime:=Heure courante
    ` Essayons de le charger en mémoire
DOCUMENT VERS BLOB(Document;$vxData)
Si (OK=1)
    ` Si le document a pu être chargé dans le BLOB,
    ` Démarrer la procédure stockée qui va importer les données
    ` sur le poste serveur
    $spProcessID:=Executer sur serveur("SERVER IMPORT";32*1024;
                                     "Serveur Import Services";Table($1);$2;$vxData)
    ` Nous n'avons alors plus besoin du BLOB dans ce process
EFFACER VARIABLE($vxData)
    ` Attendons l'achèvement de l'opération effectuée par la procédure stockée
Repete
    ENDORMIR PROCESS(Numero du process courant;300)
    LIRE VARIABLE PROCESS($spProcessID;spErrCode;spErrCode)
    Si (Indefinie(spErrCode))
        ` Note: si la procédure stockée n'a pas initialisé sa propre instance
        ` de la variable spErrCode, il se peut qu'une variable indéfinie
        ` soit retournée
        spErrCode:=1
    Fin de si
Jusque (spErrCode<=0)
    ` Envoyons un accusé de réception à la procédure stockée
    spErrCode:=1
ECRIRE VARIABLE PROCESS($spProcessID;spErrCode;spErrCode)
$vhEndTime:=Heure courante
ALERTE("L'opération a duré "+Chaine(0+($vhEndTime-$vhStartTime))+
                                               " secondes.")

Sinon
    ALERTE("Il n'y a pas assez de mémoire pour charger le document.")
Fin de si
Fin de si

```

Voici la méthode projet SERVER IMPORT exécutée en tant que procédure stockée :

```
  ` Méthode projet SERVER IMPORT
  ` SERVER IMPORT ( Entier long ; Alpha ; BLOB )
  ` SERVER IMPORT ( Numéro de table ; Formulaire entrée ; Données importées )

C_ENTIER LONG($1)
C_ALPHA(31;$2)
C_BLOB($3)
C_ENTIER LONG(spErrCode)

  ` L'opération n'est pas encore terminée, affectons 1 à spErrCode
spErrCode:=1
$vpTable:=Table($1)
FORMULAIRE ENTREE($vpTable->,$2)
$vsDocName:="Fichier Import "+Chaîne(1+Hasard)
Si(Sous Windows)
  $vsDocName:=$vsDocName+".txt" ` Sous Windows, l'extension est obligatoire
Fin de si
SUPPRIMER DOCUMENT($vsDocName)
BLOB VERS DOCUMENT($vsDocName;$3)
LECTURE ASCII($vpTable->,$vsDocName)
SUPPRIMER DOCUMENT($vsDocName)
  ` L'opération est terminée, affectons 0 à spErrCode
spErrCode:=0
  ` Attendons que le poste client à l'origine de la requête ait reçu les résultats
Repeter
  ENDORMIR PROCESS(Numero du process courant;1)
Jusque (spErrCode>0)
```

Note : La méthode projet Sous Windows est fournie dans la section Présentation des documents système du manuel Langage de 4D.

Une fois que ces deux méthodes projet ont été implémentées dans votre base, vous pouvez effectuer un import basé sur une procédure stockée, en écrivant par exemple :

```
CLIENT IMPORT (->[Table1];"Import")
```

Si vous réalisez quelques tests comparatifs, vous pourrez constater qu'avec ce type de méthode, l'import des enregistrements est jusqu'à 60 fois plus rapide qu'un import "classique".

Référence

ECRIRE VARIABLE PROCESS, Executer sur serveur, LIRE VARIABLE PROCESS, Procédures stockées, Services basés sur les procédures stockées (exemple).

Dans l'exemple présenté dans la section Import basé sur les procédures stockées (exemple), une procédure stockée est lancée puis stoppée à chaque fois qu'une opération d'import de données est requise. Dans le présent exemple, une procédure stockée est lancée automatiquement lorsque la base serveur est ouverte et peut être stoppée et redémarrée à volonté par tout 4D Client connecté à la base. Dès qu'elle est exécutée, la procédure stockée peut répondre de manière asynchrone à de multiples requêtes envoyées par les clients connectés à la base.

Alors que la section Import basé sur les procédures stockées (exemple) vous montre comment optimiser un service 4D Server existant, cet exemple vous montre comment créer des services nouveaux et personnalisés disponibles pour tous les 4D Client. En outre, cet exemple peut être utilisé comme modèle pour la création de vos propres services.

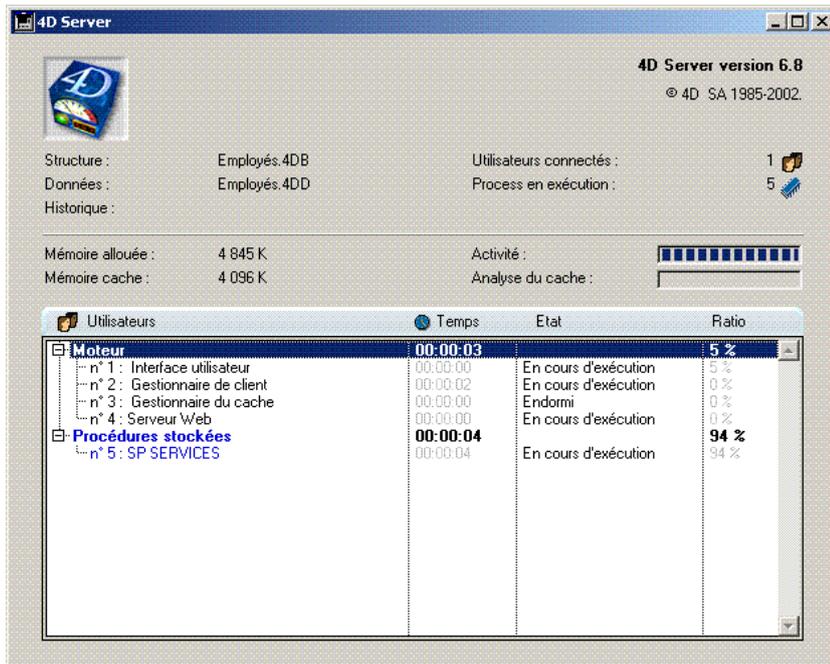
Démarrage automatique de la procédure stockée

La procédure stockée est lancée automatiquement par la Méthode base Sur démarrage serveur :

```
` Méthode base Sur démarrage serveur  
START SP SERVICES
```

Comme la Méthode base Sur démarrage serveur lance la méthode projet SP SERVICES en tant que procédure stockée, cette dernière s'exécute dès que la base est ouverte avec 4D Server, que des client soient connectés à la base ou non.

Ci-dessous, la fenêtre d'administration de 4D Server affiche la procédure stockée en cours d'exécution alors qu'aucun client n'est encore connecté.



Démarrer et stopper à volonté la procédure stockée

Voici la méthode projet START SP SERVICES :

```

` Méthode projet START SP SERVICES
<>vISPServices:=Executer sur serveur("SP SERVICES";32*1024;"SP SERVICES";*)

```

Comme la commande Executer sur serveur se comporte comme Nouveau process lorsqu'elle est appelée sur le serveur, la même méthode (START SP SERVICES) peut être utilisée sur le poste serveur ou sur tout poste client pour démarrer à volonté la méthode SP SERVICES en tant que procédure stockée sur le serveur.

La méthode projet STOP SP SERVICES "ordonne" à la méthode projet SP SERVICES de s'arrêter.

```

` Méthode projet STOP SP SERVICES
ECRIRE VARIABLE PROCESS(<>vISPServices;vbStopSPServices;Vrai)

```

La méthode projet SP SERVICES, lorsqu'elle démarre, met la variable process vbStopSPServices à Faux puis s'exécute en boucle jusqu'à ce que cette variable booléenne devienne Vraie. Grâce à la commande ECRIRE VARIABLE PROCESS, tout process utilisateur exécuté sur le serveur ou sur un poste client peut modifier la valeur de la variable vbStopSPServices, et donc stopper la procédure quand il le souhaite.

Communiquer avec la procédure stockée

La procédure stockée doit pouvoir, de manière asynchrone, recevoir des requêtes des clients et y répondre à tout moment et dans n'importe quel ordre. Le moyen le plus simple et direct d'assurer cette communication est d'utiliser une table.

SP Requests	
reqID	L
reqType	A
reqStatus	I
reqData	X
reqParams	T

La table [SP Requests] contient les champs suivants :

- [SP Requests]reqID est rempli à l'aide de la commande Numerotation automatique. Ce champ identifie chaque requête de manière unique.
- [SP Requests]reqType décrit le type de la requête.
- [SP Requests]reqStatus peut prendre une des valeurs suivantes :

Valeur	Description
1	la requête a été reçue mais pas encore traitée.
0	la requête a été traitée avec succès.
< 0	la requête a été traitée mais une erreur s'est produite.

Note : Ces valeurs ont été choisies arbitrairement pour cet exemple, elles ne sont pas issues de 4D.

- [SP Requests]reqData est un BLOB contenant les données de la requête. Il peut contenir les données envoyées par le demandeur ou les données retournées par la procédure stockée au demandeur.
- [SP Requests]reqParams contient éventuellement les valeurs de paramètres envoyés par le demandeur à la procédure stockée.

Pourquoi utiliser une table ?

La communication entre un process client et une procédure stockée peut être établie par l'intermédiaire des commandes LIRE VARIABLE PROCESS, ECRIRE VARIABLE PROCESS et VARIABLE VERS VARIABLE. C'est, par exemple, la solution retenue dans la section Import basé sur les procédures stockées (exemple) ainsi que pour la méthode projet STOP SP SERVICES listée plus haut.

Ici, le système doit permettre à la procédure stockée de recevoir et de renvoyer des quantités variables de données. On pourrait utiliser des tableaux (y compris des tableaux Texte et Image), mais il existe deux raisons principales pour préférer l'emploi d'une table :

- L'algorithme de gestion des requêtes via des enregistrements est plus simple à créer. Envoyer d'une requête depuis un poste client consiste simplement à ajouter une requête dans la table. Répondre à la requête depuis la procédure stockée consiste simplement à modifier cette requête.
- Comme les requêtes sont stockées dans une table, elles le sont sur le disque. Par conséquent, si la taille d'une requête est importante, cela ne constituera pas un problème dans la mesure où elle peut être supprimée de la mémoire (contrairement aux données stockées dans des tableaux).

Envoyer une requête depuis le poste client

La méthode projet Client post request est une méthode générique pour envoyer une requête :

```

` Méthode projet Client post request
` Client post request ( Chaîne { ; Texte } ) -> Entier long
` Client post request ( Type de requête { ; Parametres } ) -> Numéro de requête
CREER ENREGISTREMENT([SP Requests])
[SP Requests]reqID:=Numerotation automatique([SP Requests])
[SP Requests]reqType:=$1
[SP Requests]reqStatus:=1
Si (Nombre de parametres>=2)
    [SP Requests]reqParams:=$2
Fin de si
STOCKER ENREGISTREMENT([SP Requests])
$0:=[SP Requests]reqID

```

La méthode retourne le numéro de la requête, dont l'unicité est garantie par l'utilisation de la commande Numerotation automatique. Une fois que cet enregistrement a été ajouté dans la table [SP Requests], le client n'a plus qu'à interroger régulièrement le champ [SP Requets]redStatus jusqu'à ce que la procédure stockée ait terminé le traitement de la requête.

Tester le statut de la requête et récupérer le résultat sur le poste client

La méthode projet Client get result est une méthode générique pour tester le statut de la requête. Comme expliqué précédemment, dès que la valeur du champ [SP Requets]redStatus devient différente de 1, le client sait que la procédure stockée a traité (avec succès ou non) la requête.

```

    ` Méthode projet Client get result
    ` Client get result ( Entier long ; ->BLOB {Entier long } ) -> Entier long
    ` Client get result ( Numéro de requête; ->Données {; Durée } ) -> Code d'erreur
C_ENTIER LONG($0;$1;$vIDelay)
    $0:=1
    $vIDelay:=0
Si (Nombre de parametres>=3)
    $vIDelay:=$3
Fin de si
LECTURE SEULEMENT([SP Requests])
Repeter
    CHERCHER([SP Requests];[SP Requests]reqID=$1)
Si (Enregistrements trouves([SP Requests])>0)
    Si ([SP Requests]reqStatus # 1)
    $2->:=[SP Requests]reqData
    LECTURE ECRITURE([SP Requests])
    Tant que (Enregistrement verrouille([SP Requests]))
    WAITING LOOP ($vIDelay)
    CHARGER ENREGISTREMENT([SP Requests])
    Fin tant que
    SUPPRIMER ENREGISTREMENT([SP Requests])
    $0:=[SP Requests]reqStatus
    Fin de si
Sinon
    ` L'enregistrement de la requête a été perdu !
    ` Cela ne devrait pas se produire. Mais fixons tout de même
    ` le code d'erreur -2 (valeur arbitraire)
    $0:=-2
Fin de si

```

```

    ` La requête n'a pas encore été traitée
Si ($0=1)
    WAITING LOOP ($vIDelay)
Fin de si
Jusque ($0 # 1)
LECTURE SEULEMENT([SP Requests])

```

Si la requête a été traitée avec succès par la procédure stockée, la méthode copie le résultat (s'il y en a un) de l'enregistrement vers le BLOB sur lequel un pointeur a été passé en paramètre. La méthode appelante analyse puis exploite les données du BLOB en fonction du type de la requête. Notez que c'est au client de supprimer l'enregistrement de [SP Requests] une fois que la requête est terminée.

La petite méthode projet WAITING LOOP effectue une boucle pendant un certain nombre de ticks :

```

    ` Méthode projet WAITING LOOP
    ` WAITING LOOP ( Entier long )
    ` WAITING LOOP ( Durée en ticks )
C_ENTIER LONG($1)
$vStartTicks:=Nombre de ticks
Repete
    APPELER 4D
Jusque ((Nombre de ticks-$vStartTicks)>=$1)

```

Rappel : ENDORMIR PROCESS n'a pas d'effet sur le process principal (Utilisation/Menus créés). Si vous utilisez la méthode projet WAITING LOOP, le process attendra pendant le temps défini même si la requête émane du process principal d'un poste client.

La procédure stockée et ses sous-routines

La méthode projet SP SERVICES est la méthode exécutée en tant que procédure stockée sur le poste serveur. La structure générale de cette méthode, présentée ci-dessous, est très simple :

```

Initialisation d'une variable "stop"
Repete
    Recherche des requêtes dont le champ [SP Requests]reqStatus est égal à 1
    Pour chaque requête
        En fonction du type de la requête, appeler une sous-routine
            qui stocke le résultat dans le champ [SP Requests]reqData
        Changer le statut de la requête pour que le client sache ce qui s'est passé
    Fin de boucle
    "Dormir" un peu avant de recommencer
Jusqu'à ce que la variable "stop" devienne vraie

```

Voici le code source réel :

```
    ` Méthode projet SP SERVICES
    ` La procédure stockée démarre
vbStopSPServices:=Faux
    ` La procédure stockée n'a pas besoin d'accès en écriture aux tables...
LECTURE SEULEMENT(*)
    ` ...sauf pour la table [SP Requests]
LECTURE ECRITURE([SP Requests])
Repete
    ` Recherche des requêtes non encore traitées
CHERCHER([SP Requests];[SP Requests]reqStatus=1)
    ` Traitement de ces requêtes les unes après les autres
Boucle ($vIRecord;1;Enregistrements trouvés([SP Requests]))
    ` Si l'enregistrement de la requête est verrouillé, attendre son déverrouillage

Tant que (Enregistrement verrouille([SP Requests]))
    ` Attendre une seconde avant d'essayer encore
ENDORMIR PROCESS(Numero du process courant;60)
    ` Essayer de charger l'enregistrement
CHARGER ENREGISTREMENT([SP Requests])
Fin tant que
    ` Supposons que la requête sera traitée avec succès
[SP Requests]reqStatus:=0
Au cas ou
    : ([SP Requests]reqType="Server Information")
        SP DO SERVER INFORMATION
    : ([SP Requests]reqType="Volume List")
        SP DO VOLUME LIST
    : ([SP Requests]reqType="Browse Directory")
        SP DO BROWSE DIRECTORY ([SP Requests]reqParams)
    ` ...
    ` D'AUTRES TYPES DE REQUETES PEUVENT ETRE AJOUTES ICI !
    ` ...
Sinon
    ` Le type de requête est inconnu, retourner l'erreur -1
    ` (valeur arbitraire)
[SP Requests]reqStatus:=-1
Fin de cas
    ` Forcer le statut de la requête à être différent de 1
    ` (pour le cas où une sous-routine lui a donné la valeur 1)
Si ([SP Requests]reqStatus=1)
[SP Requests]reqStatus:=-3
Fin de si
```

```

    ` Mettre à jour l'enregistrement de la requête
STOCKER ENREGISTREMENT([SP Requests])
    ` Passer à la requête non traitée suivante
ENREGISTREMENT SUIVANT([SP Requests])
Fin de boucle
    ` Libérer le dernier enregistrement traité
LIBERER ENREGISTREMENT([SP Requests])
    ` Attendre une seconde avant de continuer à répondre aux requêtes
ENDORMIR PROCESS(Numero du process courant;60)
    ` Boucle jusqu'à ce qu'on ordonne à la procédure stockée
    ` de stopper son exécution
Jusque (vbStopSPServices)

```

La méthode projet SP SERVICES peut être utilisée comme modèle pour implémenter de nouveaux services dans une base. Dans ce document, nous détaillons les sous-routines SP DO SERVER INFORMATION et SP DO VOLUME LIST. La sous-routine SP DO BROWSE DIRECTORY (qui par ailleurs accepte comme paramètre le paramètre envoyé par le client dans le champ [SP Requests]reqParams) ne sera pas traitée ici.

En fonction du type de la requête, la méthode projet SP SERVICES appelle une sous-routine dont le rôle est de stocker les données résultantes dans le champ [SP Requests]reqData field. La sauvegarde de l'enregistrement et la modification du statut de la requête sont effectuées par la méthode projet SP SERVICES.

Voici la sous-routine SP DO SERVER INFORMATION. Elle stocke des informations relatives au serveur dans le BLOB. Une autre méthode projet extraira les données du BLOB en fonction du poste client.

```

    ` Méthode projet SP DO SERVER INFORMATION
TEXTE VERS BLOB(Version application(*);[SP Requests]reqData;Chaîne pascal)
TEXTE VERS BLOB(Fichier structure;[SP Requests]reqData;Chaîne pascal;)
TEXTE VERS BLOB(Fichier donnees;[SP Requests]reqData;Chaîne pascal;)
PROPRIETES PLATE FORME($vlPlatform;$vlSystem;$vlMachine)
VARIABLE VERS BLOB($vlPlatform;[SP Requests]reqData;*)
VARIABLE VERS BLOB($vlSystem;[SP Requests]reqData;*)
VARIABLE VERS BLOB($vlMachine;[SP Requests]reqData;*)

```

Voici la sous-routine SP DO VOLUME LIST. Elle stocke des informations relatives aux volumes dans le BLOB. Une autre méthode projet extraira les données du BLOB en fonction du poste client.

```

    ` Méthode projet SP DO VOLUME LIST
LISTE DES VOLUMES($asVName)
    $vISize:=Taille tableau($asVName)
TABLEAU REEL($arVSize;$vISize)
TABLEAU REEL($arVUsedSpace;$vISize)
TABLEAU REEL($arVFreeSpace;$vISize)
Boucle ($vIElem;1;$vISize)
    PROPRIETES DU VOLUME($asVName{$vIElem};$arVSize{$vIElem};
        $arVUsedSpace{$vIElem};$arVFreeSpace{$vIElem})

Fin de boucle
VARIABLE VERS BLOB($asVName;[SP Requests]reqData)
VARIABLE VERS BLOB($arVSize;[SP Requests]reqData;*)
VARIABLE VERS BLOB($arVUsedSpace;[SP Requests]reqData;*)
VARIABLE VERS BLOB($arVFreeSpace;[SP Requests]reqData;*)

```

Afficher les informations du serveur sur un poste client

Avec les méthodes projet génériques Client post request et Client get result, la méthode projet M_SERVER_INFORMATION affiche, sur le poste client, les informations sur le serveur retournées par la procédure stockée. Cette méthode pourrait être associée à une commande de menu ou appelée, par exemple, depuis la méthode objet d'un bouton :

```

    ` M_SERVER_INFORMATION
C_BLOB(vxData)
C_ENTIER LONG($vIReqID;$vIErrCode;$vIOffset)
    ` Envoi de la requête
    $vIReqID:=Client post request ("Server Information")
    ` Test du statut de la requête et réception du résultat
    $vIErrCode:=Client get result ($vIReqID;->vxData;60)
    ` Si la requête est terminée avec succès, affichage du résultat
Si ($vIErrCode=0)
    ` Extraction de l'information résultante du BLOB
    $vIOffset:=0
    vsServerVersion:=BLOB vers texte(vxData;Chaîne pascal;$vIOffset)
    vsStructureFile:=BLOB vers texte(vxData;Chaîne pascal;$vIOffset)
    vsDataFile:=BLOB vers texte(vxData;Chaîne pascal;$vIOffset)
BLOB VERS VARIABLE(vxData;$vIPlatform;$vIOffset)
BLOB VERS VARIABLE(vxData;$vISystem;$vIOffset)
BLOB VERS VARIABLE(vxData;$vIMachine;$vIOffset)

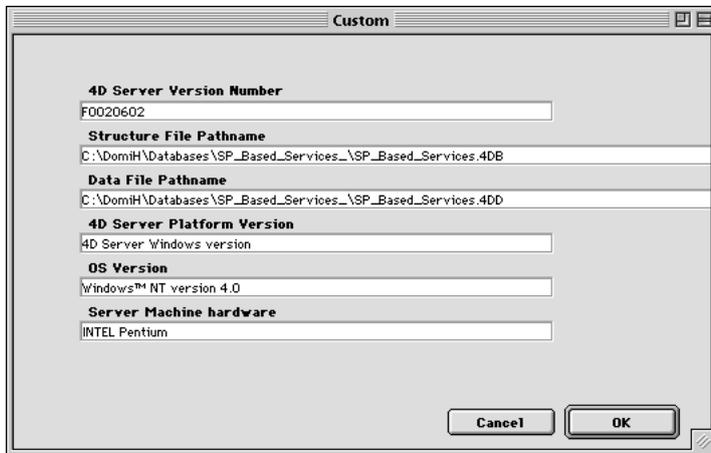
```

```

    \ Analyse des propriétés de la plate-forme
vs4DPlatform:="Version de 4D Server inconnue"
vsSystem:="Version du système inconnue"
vsMachine:="Ordinateur inconnu"
    \ ...
    \ Ici est placé le code (non listé) qui analyse $vlSystem et $vlMachine
    \ ( reportez-vous à l'exemple de la commande PROPRIETES PLATE FORME)
    \ ...
    \ Affichage de l'information résultante
DIALOGUE([SP Requests];"SERVER INFORMATION")
Si
    ALERTE("Erreur de requête "+Chaine($vlErrCode))
Fin de si
    \ Le BLOB est désormais inutile
EFFACER VARIABLE(vxData)

```

Voici le formulaire [SP Requests];"SERVER INFORMATION" en mode Utilisation ou Menus créés :



Dans cette fenêtre, l'information provient d'un 4D Server installé sur une machine Windows NT et est affichée sur un client Macintosh.

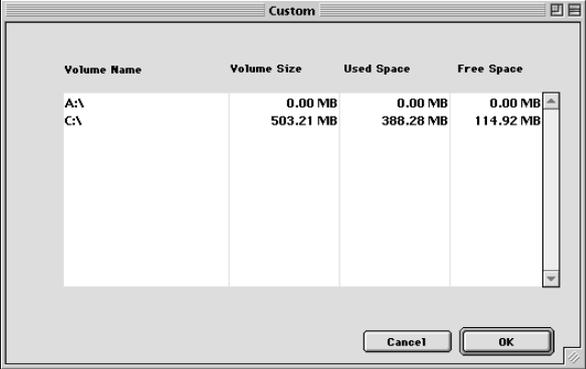
Afficher la liste des volumes du serveur sur un poste client

Avec les méthodes projet génériques Client post request et Client get result, la méthode projet M_SERVER_VOLUMES affiche, sur le poste client, les informations sur la liste des volumes du serveur retournée par la procédure stockée.

Cette méthode pourrait être associée à une commande de menu ou appelée, par exemple, depuis la méthode objet d'un bouton :

```
    ` M_SERVER_VOLUMES
C_BLOB(vxData)
    ` Envoi de la requête
    $vReqID:=Client post request ("Volume List")
    ` Test du statut de la requête et réception du résultat
    $vErrCode:=Client get result ($vReqID;->vxData;120)
    ` Si la requête est terminée avec succès, affichage du résultat
Si ($vErrCode=0)
    ` Extraction de l'information résultante du BLOB
    $vOffset:=0
    BLOB VERS VARIABLE(vxData;asVName;$vOffset)
    BLOB VERS VARIABLE(vxData;arVSize;$vOffset)
    BLOB VERS VARIABLE(vxData;arVUsedSpace;$vOffset)
    BLOB VERS VARIABLE(vxData;arVFreeSpace;$vOffset)
    Boucle ($vElem;1;Taille tableau(arVSize))
        ` Conversion d'octets en méga-octets
        arVSize{$vElem}:=arVSize{$vElem}/1048576
        arVUsedSpace{$vElem}:=arVUsedSpace{$vElem}/1048576
        arVFreeSpace{$vElem}:=arVFreeSpace{$vElem}/1048576
    Fin de boucle
        ` Affichage de l'information résultante
    DIALOGUE([SP Requests];"VOLUME LIST")
Sinon
    ALERTE("Erreur de requête "+Chaine($vErrCode))
Fin de si
    ` Le BLOB est désormais inutile
EFFACER VARIABLE(vxData)
```

Voici le formulaire [SP Requests];"VOLUME LIST" en mode Utilisation ou Menus créés :



The screenshot shows a window titled "Custom" with a table of volume information. The table has four columns: "Volume Name", "Volume Size", "Used Space", and "Free Space". There are two rows of data: "A:\\" and "C:\\". The "C:\\" row shows a volume size of 503.21 MB, 388.28 MB of used space, and 114.92 MB of free space. The "A:\\" row shows 0.00 MB for all three metrics. The window has "Cancel" and "OK" buttons at the bottom.

Volume Name	Volume Size	Used Space	Free Space
A:\	0.00 MB	0.00 MB	0.00 MB
C:\	503.21 MB	388.28 MB	114.92 MB

Dans cette fenêtre, l'information provient d'un 4D Server installé sur une machine Windows et est affichée sur un client Macintosh.

Référence

Commandes du thème BLOB, Executer sur serveur, Import basé sur les procédures stockées (exemple), Procédures stockées.